1977-03

Emulation of the AN/UYK-20 tactical data computer on the Burroughs D-machine

Anzelmo, Ralph Harry; Kaye, Theodore Lawrence

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

EMULATION OF THE AN/UYK-20
TACTICAL DATA COMPUTER
ON THE BURROUGHS D-MACHINE

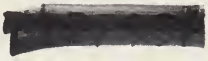
by

Ralph Harry Anzelmo
and
Theodore Lawrence Kaye

March 1977

Thesis Advisor: L. V. Rich

Approved for public release; distribution unlimited.



REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

Emulation of the AN/UYK-20 Tactical Data Computer on the Burroughs D-machine

5. TYPE OF REPORT & PERIOD COVERED

Master's Thesis; March 1977

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Ralph Harry Anzelmo and
Theodore Lawrence Kaye

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

March 1977

13. NUMBER OF PAGES

263

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

Naval Postgraduate School
Monterey, California 93940

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

emulation
microprogramming
AN/UYK-20
Burroughs D-machine

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A representation of the Univac AN/UYK-20 computer systems has been emulated on the Burroughs D Interpreter-Based System. The entire AN/UYK-20 instruction repertoire has been emulated with the exception of the "math pac" option (floating point arithmetic and cordic functions), clock and interrupt codes, and input/output operation codes. Modular design with extensive documentation has been implemented throughout program (cont.)

20. (cont.)

development allowing for ease of modification and further extensions to the existing emulation. Emulation and hardware architecture of the AN/UYK-20 and the Burroughs D-machine, are discussed in conjunction with the AN/UYK-20 itself. Methods of testing and debugging, sample test programs and recommendations for continued design modifications to the emulation are presented.

Approved for public release; distribution unlimited.

Emulation of the AN/UYK-20 Tactical Data Computer on the Burroughs D-machine

by

Ralph Harry Anzelmo
Captain, United States Marine Corps
B.A., Montclair State College, 1968

Theodore Lawrence Kaye
Lieutenant, United States Navy
B.S.S.E., United States Naval Academy, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
MARCH 1977

ABSTRACT

A representation of the Univac AN/UYK-20 computer system has been emulated on the Burroughs D Interpreter-Based System. The entire AN/UYK-20 instruction repertoire has been emulated with the exception of the 'math pac' option (floating point arithmetic and cordic functions), clock and interrupt codes, and input/output operation codes. Modular design with extensive documentation has been implemented throughout program development allowing for ease of modification and further extensions to the existing emulation. Emulation and the hardware architecture of the AN/UYK-20 and the Burroughs D-machine, are discussed in conjunction with the AN/UYK-20 emulation itself. Methods of testing and debugging, sample test programs and recommendations for continued design modifications to the emulation are presented.

CONTENTS

I. INTRODUCTION..... 9

A. STATEMENT OF THE PROBLEM..... 9

B. APPLICATIONS OF THE AN/UYK-20 10

C. PROJECT DESIGN OBJECTIVES..... 11

II. EMULATION..... 13

A. HISTORICAL BACKGROUND..... 13

B. MICROPROGRAMMING..... 16

C. THE GOALS OF EMULATION..... 21

D. EMULATION VERSUS SIMULATION..... 21

E. EMULATION TECHNIQUES..... 23

F. EMULATION HARDWARE..... 25

III. AN/UYK-20 ARCHITECTURE..... 28

A. HARDWARE DESIGN..... 29

B. INSTRUCTION FORMATS AND REPERTOIRE..... 38

1. Repertoire of Instructions..... 38

2. Instruction Format..... 41

IV. BURROUGHS D-MACHINE..... 46

A. HARDWARE DESCRIPTION..... 46

1. Logic Unit..... 48

2. The Control Unit..... 52

3. Memory Control Unit..... 53

4. Microprogram memory (M-memory)..... 55

B. NPS MICROPROGRAMMING FACILITY..... 56

1. Physical Description..... 56

2. Input/Output Interface..... 58
3. Memory Interface..... 59
C. MICROINSTRUCTION TIMING..... 60
D. TRANSLANG..... 64

V. AN/UYK-20 EMULATOR..... 66
A. EMULATION DESIGN..... 66
1. Functional Components..... 66
2. Main Memory Organization..... 68
3. Emulation Program Status Word..... 70
4. D-Machine Registers..... 73
B. LOADER..... 75
C. THE FETCH MODULE..... 77
D. OPCODE IMPLEMENTATION..... 83
E. UTILITIES..... 86
F. INPUT/OUTPUT CONTROLLER..... 87

VI. EMULATION TESTING..... 91
A. METHOD OF TESTING..... 91
B. SAMPLE TEST PROGRAMS..... 94
C. TEST RESULTS..... 95

VII. SUMMARY AND RECOMMENDATIONS..... 97
A. EXPERIENCE WITH HARDWARE..... 97
B. LESSONS LEARNED..... 98
C. EMULATION PROBLEMS..... 99
D. RESULTS.....100
E. RECOMMENDATIONS AND FOLLOW-ON TOPICS 101

APPENDIX A. AN/UYK-20 EMULATOR USER'S MANUAL.....103

APPENDIX B. LOADER CONTROL CARD FORMATS.....107

APPENDIX C. AN/UYK-20 EMULATOR INSTRUCTION FORMAT.....110

APPENDIX D. SAMPLE DEBUGGER OUTPUT.....111

APPENDIX E. SAMPLE TEST PROGRAMS.....113

APPENDIX F. EMULATOR LISTING.....120

BIBLIOGRAPHY.....259

INITIAL DISTRIBUTION LIST.....262

ACKNOWLEDGEMENTS

Our sincere gratitude is expressed to Lieutenant Lyle V. Rich, SC, USN for sponsoring this thesis and without whose guidance this research would not have been a success. Technical expertise for the AN/UYK-20 was provided by John H. Westergren, Field Engineer, Mini-Systems Support Operations, Sperry Univac Computer Systems, St. Paul, Minnesota. Initial instruction on the operation and design of the Burroughs D-machine was graciously provided by Lieutenant Jerry M. Haggerty, USN and Lieutenant John M. Hartling, USN. Finally, the authors would like to dedicate this thesis to their wives, whose love, devotion, and understanding enabled this project to be completed.

I. INTRODUCTION

A. STATEMENT OF THE PROBLEM

The Navy has been challenged with maintaining the newest, most efficient tactical data systems consistent with the continually increasing demands and requirements of the real-time environment. There is an extensive conversion effort required to change from existing systems to newer more sophisticated technology such as the AN/UYK-20. Inherent in upgrading to a new system is the complex software redesign and modification process which is often hindered by the absence of the new computer system.

Unfortunately, the demands of a military installation require software generation prior to implementation of an upgraded computer system. One solution to this problem is to utilize for software development an intermediate computer system which has the capability of emulating the anticipated target machine. This provides a vehicle for software design, development, and testing prior to transitioning to the new system.

Currently, the Naval Postgraduate School (NPS) Computer Science Department maintains a Burroughs Interpreter-Based System known as the Burroughs D-machine. The D-machine is capable of being microprogrammed to emulate any of a miriad

of target machines. It effectively enables students to create their own computer knowing only the machine instruction repertoire for the control unit in the target machine.

The problem presented was to develop a feasible working model of the AN/UYK-20 on the microprogrammable Burroughs D-machine. The project provided an opportunity to obtain practical experience with contemporary hardware and to manipulate writeable control stores to imitate a Navy tactical computer.

B. APPLICATIONS OF THE AN/UYK-20

The Univac AN/UYK-20 minicomputer is a general purpose militarized digital computer adaptable to numerous tactical applications. The AN/UYK-20 has been successfully utilized in many time-critical, real-time systems including fire control radar, communication controllers, signal processing analyzers for sonar and beacon signals, and numerous weapons control systems. A subsequent chapter will be devoted to the technical aspects and internal design of the AN/UYK-20.

Because of its size, ruggedness, and computing capabilities, the AN/UYK-20 has been designated the Navy's standard tactical minicomputer [16]. It was selected for emulation in order to provide a feasible platform for software development to those military installations either contemplating or in the process of receiving an AN/UYK-20.

A workable emulation would allow military applications such as data reduction, navigation, telemetry, sensor processing, range tracking and logistics to have software packages developed, tested, and modified prior to arrival of the AN/UYK-20. Furthermore, it would permit personnel to become familiar with the machine by providing advanced training, thereby easing the transition phase to the new system.

C. PROJECT DESIGN OBJECTIVES

Several design techniques were used throughout the development of this project: 1) modularity, 2) structured programming, and 3) extensive documentation. These design features will aid the interested reader as well as simplify any future extensions or modifications to the existing emulation.

Modular design was utilized by creating independent program segments which were individually developed, debugged, and tested. These modules or subroutines provided a strong foundation which were readily modified throughout the entire programming effort. Conceptually, the emulation was divided into relatively small entities which were further reduced to program segments rarely exceeding one page in length.

Structured programming was demonstrated by utilizing a limited number of control flow structures and maintaining a common logical design throughout the entire emulation. Comprehensible code held precedence over extremely efficient

code.

In addition to modularity and structured programming, the entire programming endeavor was supplemented with extensive commenting to provide the necessary self-documentation to promote and facilitate program translation, modification, and fusing with the other independent modules. These concepts promoted the extensive team effort required to achieve the research goals. In addition, it will provide ease of program maintenance and modification in the future.

II. EMULATION

A. HISTORICAL BACKGROUND

The term microprogramming was first utilized in an article by Professor M. V. Wilkes of the Cambridge University Mathematical Laboratory in 1951 [24]. His paper concentrated on a control section within the computer which, when programmatically controlled, performed register-to-register data transfers sequentially and in parallel for the execution of a single machine instruction. A sequence of operations (microinstructions) required for execution of a machine instruction is considered a microprogram.

Traditionally, the computer has been composed of essentially five components: the arithmetic/logic unit, the control unit, memory or storage, input, and output (Figure II-1). The control section sets the proper conditions for the opening and closing of required gates in the logic network. Historically, the control section has been hardware consisting of a series of decoders and flip-flops along with their associated circuitry. Therefore, every machine instruction had a fixed interpretation which was hardwired within the control unit.

In 1957, Wilke's definition of microprogramming was slightly modified. It was defined as a technique of design-

ing the control circuits of an electronic digital computer to interpret and execute a given set of machine operations as an equivalent set of micro-operations [15].

The hardwired control section can be modified by interchanging ROM modules or other hardware components, by replacing the control section with a programmable (dynamically writeable) control store which in itself is a separate word-organized memory (Figure II-2) or by combining both approaches. A programmable control store allows rapid changes in the machine's instruction repertoire while maintaining maximum design flexibility. The resulting computer system is microprogrammable and capable of storing a series of changeable machine personalities.

The computer control store can thus be modified to allow the execution of machine language programs intended for a variety of machine architectures. This process can be compared to replacing hardware components found in conventionally designed computer systems. The primary advantage of microprogrammed logic is the capability to perform various control sequences without hardware modifications. The process through which the hardware components of one machine (host) are made to imitate the specific hardware characteristics of another machine (target) is known as emulation.

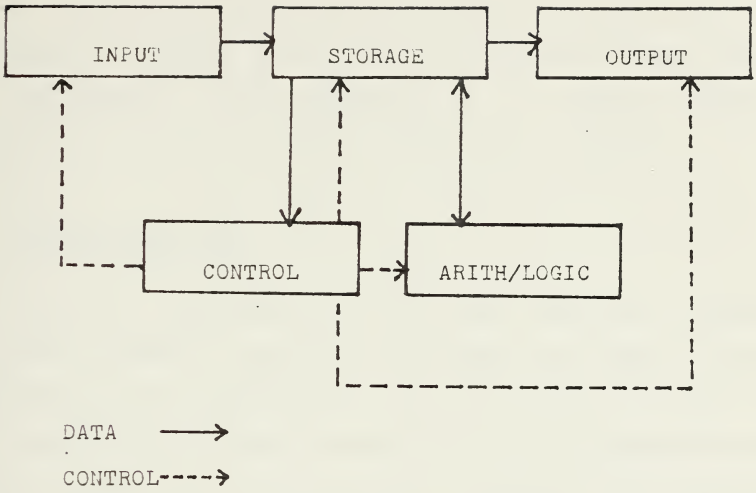


Figure II-1 [11]

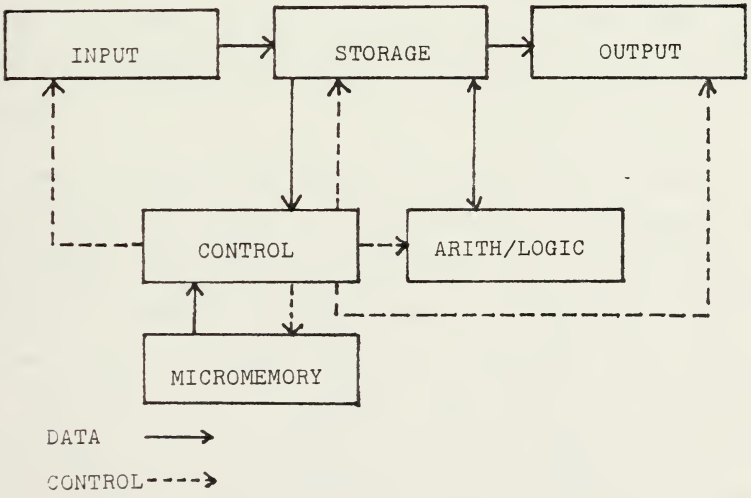


Figure II-2

Emulation allows the computer scientist to create various machine architectures from a single microprogrammable host. The complete set of microprograms (firmware) and the necessary hardware, as well as the required software, added to one computer system enabling it to execute programs designed for another system is known as an emulator.

B. MICROPROGRAMMING

Computer manufacturers have made available numerous microprogrammable machines which permit the user to tailor his instruction repertoire to meet the needs of his particular application. Some examples of microprogrammable computer systems are the Burroughs D-machine, the Nanodata QM-1, the Varian 73, the Standard Logic CASH-8, or the Hewlett-Packard 2100. These microprogrammable systems provide the benefits of flexibility, lower system costs and a systematic approach to system design if utilized effectively.

When a manufacturer designs a dynamically writeable control store, the amount of parallelism to be allowed must be determined. Parallelism is defined to be the simultaneous control of numerous hardware resources. There are basically three forms of control: vertical, horizontal, and residual. In vertical microprogramming, each instruction controls a single operation with program flow being sequential, unless the instruction was a conditional or unconditional branch. By contrast, a horizontally microprogrammed machine is

programmed via instructions which simultaneously control multiple resources including condition testing and microinstruction sequencing.

Horizontal microinstructions usually are not encoded which means each bit controls one machine resource or operation. They usually have a wider word than vertical instructions and consequently consume more memory. Vertical instructions are usually encoded with one or two levels. Encoding means the value of a control field in the microinstruction is a binary code specifying which resource or operation is to be performed. The horizontal microinstructions have the potential of being much more efficient resource managers and consequently are more difficult to optimally design than their vertical counterparts.

Combining the attributes of horizontal and vertical microprogramming results in residual control. This method saves memory by using vertical microinstructions while simultaneously controlling multiple parallel resources via setup registers.

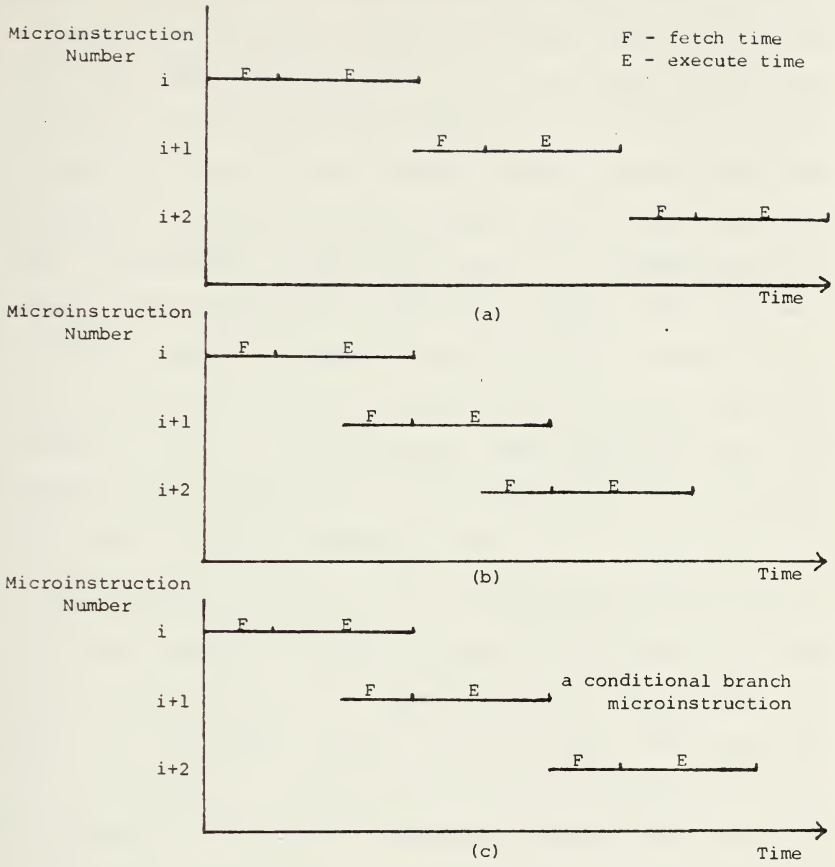
Microinstruction implementation severely effects the speed of microprogram execution. In serial implementation, one microinstruction is fetched and fully executed prior to fetching the next instruction. This technique offers the advantage of logical simplicity while suffering from lack of efficiency since it consumes the maximum amount of time.

'Parallel' implementation permits fetching of the next

instruction before termination of the previous instruction. The obvious advantage is execution speed which is of utmost importance when emulating another machine (Figure II-3) [2].

Another significant microprogramming characteristic is the number of phases used in the execution of each microinstruction. A monophase system means there are no subdivisions of the basic clock pulse and consequently each microinstruction is controlled by the transmission of the leading edge of a clock cycle. In a polyphase implementation scheme, the basic clock cycle is subdivided into minor phases which are independently generated via hardware. Although polyphase operations are more complex and require complicated control, they do permit faster resource manipulation when they are efficiently coded by allowing multiple operations to be performed during the same phase(s).

The microprogrammability of a given computer and the capabilities of its associated microprogramming language are directly effected by the the presence or absence of each of the alternative microprogramming characteristics described above. The microprogramming language spectrum ranges from the lowest level or microlanguage through the assembly languages to the high level procedural languages.



(a) Serial fetch and execute.

(b) Parallel fetch and execute.

(c) Combined serial-parallel where next address depends on conditions in present cycle.

Figure II-3 [2]

The problems of microprogramming can be significantly reduced if suitable software support exists and is readily available. This support is usually in the form of simulators and debuggers. Typically, a simulator provides an alternative to assembly level coding by permitting the user to code in a higher level language and yet achieve the same results at the expense of some added memory and execution time. Debuggers are extremely useful in the developmental stages of microprogramming especially for new and experimental system design. Debuggers permit dynamic access to the machine status and register contents at the instant they are employed, i.e. a trace feature. Some debuggers offer the opportunity of assembling in-line. This option can drastically reduce required debugging time.

The primary application of microprogramming is to implement the necessary control structure required for the analysis and execution of machine level instructions by means of programmed control stores rather than hardwired logic. Therefore, a dynamically microprogrammable computer can provide a software development system which can be a cost-effective approach to experimentation with potential candidates for replacement computer systems or the design of completely new systems to fit the needs of unique applications.

C. THE GOALS OF EMULATION

A well-designed emulation can provide an opportunity to experiment and create software for new computer systems before the actual hardware is available. The utilization of an emulator can almost eliminate reprogramming, consequently smoothing the system transition period. In addition, emulation has provided a workable model of new systems under consideration for procurement, providing a much more detailed cost-benefit analysis of system conversion.

Furthermore, it is often economically sound to emulate a second generation computer with a third generation system. This provides growth to a contemporary system while fulfilling the requirements of the past in a cost-effective manner. However, this can be a disadvantage if the programming staff uses the emulation as a link to the old system and consequently fails to take advantage of the attributes of the new system.

D. EMULATION VERSUS SIMULATION

To accomplish the emulation objectives, certain design features must be incorporated into an emulation. Naturally, execution time and allocated memory are the two foremost considerations. Traditionally, the concept of mimicking another computer has been accomplished by either a simulator or an emulator, two concepts often confused with one another.

A simulator is a series of high level language (HLL) or assembly language statements which individually do not behave like the target machine instructions. The host machine executes its own native instructions in order to imitate the target machine operations. Consequently, simulation is a rather slow technique because it requires an intermediate translation. In addition, simulation of certain instructions such as bit manipulation and shifting operations can require an enormous amount of intermediate code generation demanding a significantly larger memory allocation.

An emulator is a microprogram that is executed on the host machine, performing machine instructions of the target machine. Since an emulator accepts the binary object code of the target machine and directly executes these instructions, it can be extremely efficient in terms of time and space requirements. The execution time of an emulation is dependent upon many factors: clock rates of the two machines, frequency of memory references, high speed shifting compatibility, required register mapping between target and host machines, bit manipulation capability of the two machines, condition code selection and testing, flexible data path selection capability, interrupt similaritites, input/output compatibility, and microprogramming efficiency. If the hardware features between target and host machines are extremely compatible and highly efficient microprogramming has been employed, an emulation performance ratio (host

to target) of nearly one to one can be attained. This emulation performance ratio (EPR) has been demonstrated by the emulation of the SKC-2070 on the Nanodata QM-1 computer. It is possible to achieve an EPR better than one to one under ideal situations, when the host machine has a much faster internal operation execution rate [1].

Several distinct advantages can be realized using emulation as compared to simulation. The execution speed is significantly better by at least an order of magnitude. The target machine representation in firmware is closer to the actual hardware design and total access to the lowest machine level is achievable. Perhaps most noteworthy, emulation provides the opportunity to rapidly create test beds for numerous machine architectures and provide a basis for new system development.

E. EMULATION TECHNIQUES

Traditionally, there have been three approaches for emulating machine instructions: 1) hardware or firmware assistance to a software simulation as demonstrated by the IBM 360/65 emulation of the IBM 7090, 2) independent host system hardware or firmware which provides for complete execution of the target machine's instruction repertoire of which the Burroughs D-machine emulation of the AN/UYK-20 is an example, and 3) an auxiliary processor which is operated in conjunction with the host machine to execute target machine instructions [14].

Software-controlled emulation is usually characterized by categorizing the target machine instructions into three distinct classes: easily emulated instructions, complex instructions not readily emulated, and those instructions not deemed necessary for the desired application. Instruction usage is significant in this classification process. Each class of instructions becomes a candidate for direct hardware or firmware implementation. The first emulated function in this approach is usually the fetch and analysis operation. After the instruction is analyzed, the appropriate opcode subfunction can be executed.

An alternative emulation technique is the firmware-controlled method. This approach is identified by having system control reside completely in firmware or hardware during the emulation process. All instructions are executed on the host machine as if they were indigenous to the target machine. This method is much more efficient than the software-controlled technique; however, it is more expensive and the cost differential is directly related to the required performance level. Performance is dependent upon the number of required data paths, arithmetic units, and other additional logic circuitry which must supplement the host machine architecture.

Upon entering the emulation mode in a firmware-controlled emulator, the machine performs like the target machine until encountering an exit situation. There exists three exit modes: 1) priority interrupt, 2) not implemented

instruction, and 3) deliberate exit because of a debugging routine.

The third emulation technique consists of utilizing auxiliary hardware electronically attached to the host computer for the sole purpose of executing target machine instructions. In effect, a target machine is composed of host machine hardware with the necessary additional components required to create an effective emulator.

F. EMULATION HARDWARE

The development of writeable control stores and microprogramming techniques have significantly influenced computer design. This section will describe some of the available dynamically microprogrammable hardware (Figure II-4).

The Hewlett-Packard 2100 is a general purpose minicomputer. It has a unique control store divided into two segments. One section is ROM and the other section is user programmable. The machine is vertically microprogrammed using a standard 80 instruction machine language repertoire. A debugger and assembler assist the user in microprogram development [2].

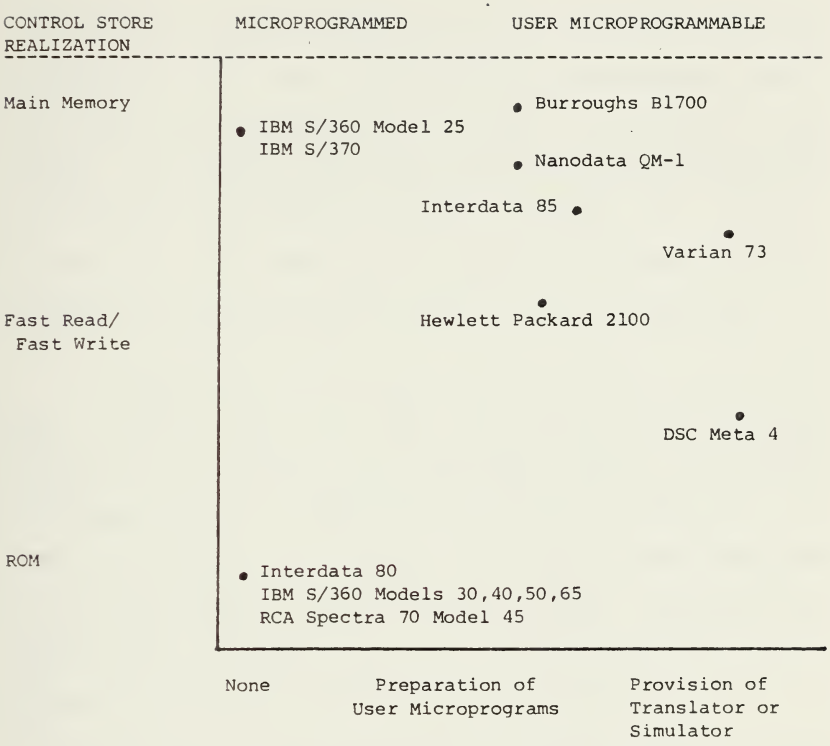
The Standard Logic CASH-8 is a high speed digital controller with a separate control store. It consists of 16 general purpose registers and an accumulator. The CASH-8 is vertically microprogrammed but does not support any language

above microlanguage [2].

The Varian 73 is a general purpose minicomputer that has a 150 instruction set. The horizontal microinstruction consists of 64 bits with 25 fields, some of which indicate register transfers, ALU operations, shifting, control store addressing, condition testing, I/O control and memory operations. The Varian 73 contains both a ROM control store and a writeable control store loadable from main memory. A microprogram assembler and interactive simulator are available [2].

The Nanodata QM-1 is unique in that it contains both a control store and a nanostore which are both loaded under user program control. The 18-bit vertical microinstructions are stored in the control store, fetched and then interpreted under nanoprogram control. A horizontal nanoinstruction is 360 bits which is subdivided into five 72-bit vectors. Assemblers for both microprograms and nanoprograms are available [2].

The previously described machines represent a small sample of the available microprogrammable computer architectures. The availability and flexibility of these computer systems has stimulated demand for these devices. Consequently, hardware manufacturers have been compelled to produce writeable control store equipment to satiate the needs of the computer market.



Note: Relative microprogrammability is the distance from the origin to the machine point in two space.

Figure II-4 [2]

III. AN/UYK-20 ARCHITECTURE

Constructing an efficient emulation requires a precise understanding of the architecture and performance characteristics of the machine being emulated. An emulation must attempt to match the target machine's features and maintain its flexibility of hardware design as closely as possible. Although it is not required, an operational demonstration of the emulated machine can solve many emulation questions.

In emulating the AN/UYK-20, architecture and performance criteria were derived from technical publications, since an actual machine was unavailable. When inconsistencies appeared in the documentation, specific questions were posed to a UNIVAC field engineer, who often tested programs on the AN/UYK-20 to resolve inconsistencies. Documentation coupled with an expert consultant provided sufficient information for emulating the AN/UYK-20 successfully.

The intent of this chapter is to outline those features of the AN/UYK-20 significant to the emulation. A detailed hardware description can be found in Refs. 20, 22, 28.

A. HARDWARE DESIGN

The AN/UYK-20 was designed for the Navy to fulfill the requirements for small or medium size general purpose data processing in shipboard, mobile shelter, or other military environments. Sperry Univac incorporated minicomputer technology in constructing the AN/UYK-20, including MSI circuitry design, microprogrammed control, memory modularity, and asynchronous or synchronous input/output channels.

The AN/UYK-20 had to be extremely flexible in its applications, offering a wide range of configuration possibilities which were derivatives of the basic design. Modularity, a concept highly desirable in a military environment, was achieved by offering options that could be easily added using printed circuit cards and/or memory modules.

The AN/UYK-20 can accomodate up to eight 8K, sixteen-bit word boards of magnetic core storage with an access time of 750 nanoseconds. The central processor is controlled by a programmable micromemory which can be expanded by an additional 512 words. The microprogram controller is programmed at the factory, but the additional micromemory option is user defined. Both sections of micromemory are programmed using fusible links, and once programmed they are completely static (Figure III-1).

A memory interface is responsible for the transfer of data to memory from the central processor (CP) and input/output controller (IOC). Both the IOC and the CP are

capable of accessing all of memory (65,536 words maximum). The addition of direct memory access (DMA) provides a second memory interface and an additional access port which is connected to each of the two 32K memory segments.

The input/output controller permits the central processor to communicate with the external devices without interfering with program execution. The IOC has a maximum of 16 parallel or serial channels. Parallel data transfer takes place asynchronously using 8-bit, 16-bit, or 32-bit transfers. Serial interfaces are either synchronous or asynchronous, with word-to-serial or serial-to-word conversions occurring in the IOC. The IOC and CP compete for memory access through the memory interface with priority given to the IOC in the event of a simultaneous request. The IOC is permanently assigned several memory addresses for command word and interrupt word storage.

The addressable high-speed registers available in the AN/UYK-20 include the program address register (P-register), two 16-bit status registers (SR1 and SR2), a real-time clock register (32-bits), a monitor clock register (16-bits), and a set of sixteen 16-bit general registers. An additional stack of 16 general registers is an available hardware option.

The sixteen general registers were included to enhance the speed and performance of the AN/UYK-20, allowing most programs to use a great proportion of register-to-register

instructions. These general registers can be used as accumulators for arithmetic, shift, or logical functions, as index registers, or as temporary storage locations. The second set of general registers can be readily employed via a status bit. This status bit designates which general register stack is to be utilized. The duplicate set of general registers yields dividends in a multi-task or heavy-interrupt processing environment. This additional register set can be used to provide high-speed temporary storage, thus avoiding slower main memory storage of working variables.

The two 16-bit status registers and the program address register represent the machine status of the AN/UYK-20. When these registers are collectively referenced, they are called the program status word (48-bit PSW). The P-register indicates the next instruction to be executed. This instruction may be a 16-bit single-word instruction or a 32-bit double-word instruction. Program control can be modified by using an instruction which manipulates the contents of the P-register.

Status register 1 contains bit information concerning condition code settings, overflow, and carry bits, interrupt codes, and numerous other machine indications (Figure III-2).

AN/UYK-20 FUNCTIONAL ARCHITECTURE

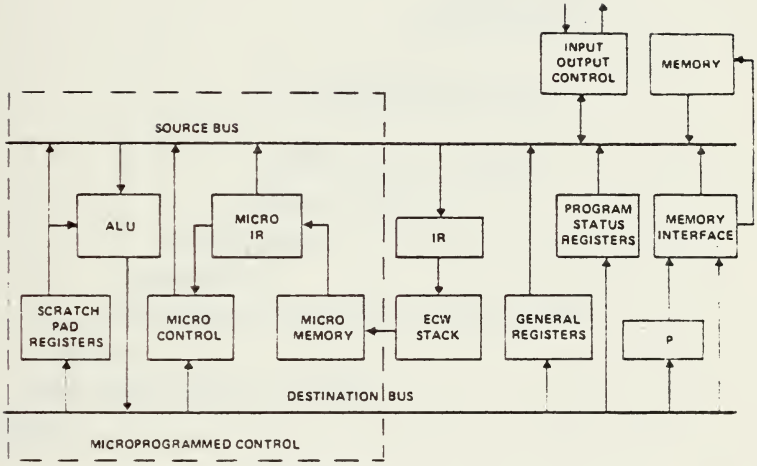
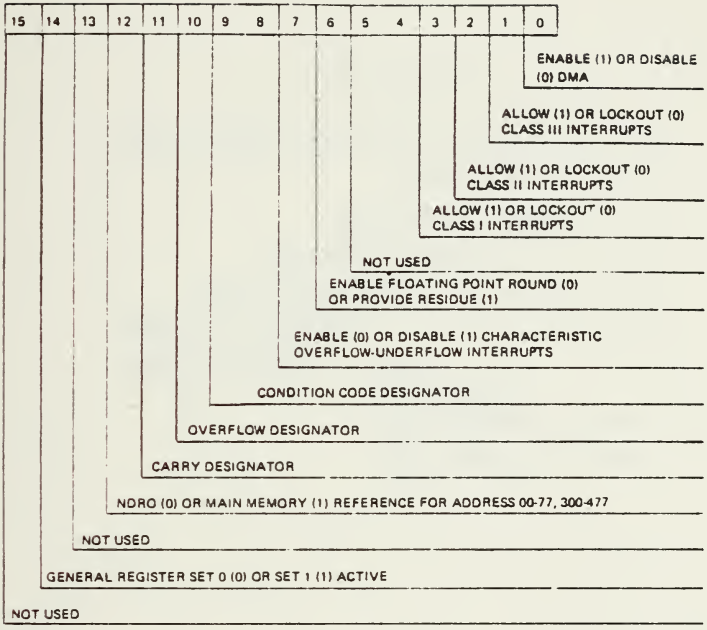


Figure III-1 [22]

STATUS REGISTER 1



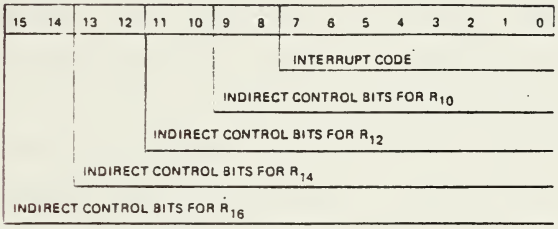
CONDITION CODES

SR BITS 8 and 9	ARITHMETIC	COMPARE
00	0	$(R_a) = (R_m)$ or (Y)
01	>0 (POS)	$(R_a) > (R_m)$ or (Y)
10	Not Used	Not Used
11	<0 (NEG)	$(R_a) < (R_m)$ or (Y)

Figure III-2 [23]

Status register 2 holds control bits for direct or indirect addressing, and holds interrupt codes. Interrupt processing routines set bits in the interrupt code field corresponding to the IOC interrupt (Figure III-3).

STATUS REGISTER 2



INDIRECT CONTROL BITS	MEANING
00	Normal Addressing.
01	Normal Addressing.
10	Indirect Addressing w/o indexing; IW1 at Y = y.
11	Indirect Addressing with indexing; IW1 at Y = y + (Rm).

Figure III-3 [23]

The real-time clock and monitor clock registers provide program-controlled interrupt capability which is useful for timing and synchronizing program segments with real-time events. The real-time clock (RTC) is a 32-bit register used as count-up storage while the monitor clock (MON) is a 16-bit count-down register. A one kHz internal oscillator controls the counting speed of both registers. An optional

external clock operating at a frequency up to 50 kHz is also available.

Interrupt processing in the AN/UYK-20 is conducted using a priority level scheme which classifies interrupts into three priority levels (classes). Interrupts within the same class are assigned a priority ranking and a code which identifies which processing routine to execute. During interrupt handling, all interrupts of the same level or lower level are locked out until the CP is completed processing the current interrupt. Higher priority interrupts can override the lockout and cause the CP to honor them first, holding the lower level interrupts in abeyance until higher level interrupt processing is completed. The highest priority interrupts are hardware malfunctions, followed by software interrupts, and at the lowest level, IOC interrupts.

Permanent locations in memory corresponding to each interrupt class hold the PSW and RTC when an interrupt is being honored. Likewise, other permanent memory addresses assigned to each interrupt class hold the appropriate interrupt routine entrance location to be loaded into the PSW.

Memory addressing is accomplished using 64 page address registers which separate memory into 1024-word pages. Absolute addresses are formed by isolating the upper six bits of the relative address to find the page address register number, and then concatenating the lower six bits of the

page address register contents with the lower ten bits of the relative address (Figure III-4). Any operation that stores into memory sets the most significant bit of the page address register used in generating the address.

Some additional hardware features of the AN/UYK-20 are those functions available on the maintenance panel of the machine itself. These include a breakpoint feature which allows an operator to insert from the panel an address which causes the AN/UYK-20 to stop execution when the selected address is referenced. Other available toggles allow halting execution programmatically using Key 1 or Key 2 on the maintenance panel. These additional hardware features are useful debugging tools.

MEMORY ADDRESS GENERATION

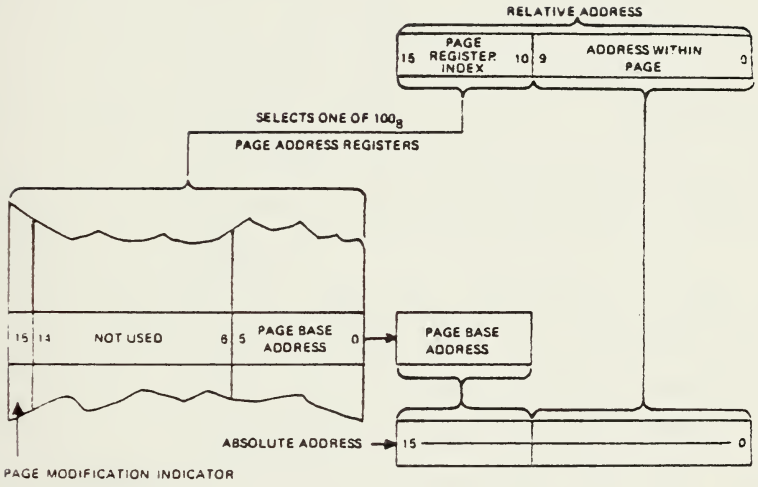


Figure III-4 [22]

B. INSTRUCTION FORMATS AND REPERTOIRE

1. Repertoire of Instructions

The AN/UYK-20 instruction set is composed of nearly 260 separate instructions designed to be both versatile and comprehensive. Both single-word (16-bit) and double-word (32-bit) instructions are available. Some of these instructions are specifically designed to meet the requirements of a real-time environment. A few sample instructions include:

a. Local jump - used to facilitate loops, saving several steps in program execution.

b. Reverse register - used in a communication environment when data is received in one sequence but must be transmitted in reverse sequence.

c. Set bit, clear bit, and test bit - used to test individual bits in registers saving considerable execution time in programs that communicate via flags and status words.

Additional flexibility is provided when the 'math pac' hardware option is included in the AN/UYK-20 configuration. Some 33 additional opcodes are added to the instruction set in order to increase the computational capabilities of the machine. An instruction for square root, double precision multiply/divide instructions, as well as hardware trigonometric and hyperbolic functions which utilize coordinate conversion algorithms (cordic) are included.

Single-word instructions are generally employed when manipulating operands in high-speed registers. Double-word instructions are used in operations involving direct or indirect addressing with or without indexing, or supplying 16-bit constants to programs. Typical instruction speeds for a single-word versus a double-word instruction are:

	SINGLE-WORD	DOUBLE-WORD
ADD	.75 - 1.5 usec	1.5 - 2.25 usec
LOAD	.75 - 1.5 usec	1.5 - 2.25 usec
MULTIPLY	3.8 - 4.0 usec	4.4 - 4.6 usec
DIVIDE	6.8 - 7.0 usec	7.4 - 7.5 usec

Nearly all instructions affect condition bits in status register 1. The AN/UYK-20 sets these bits as a result of two types of operations. Most instructions that do not involve compare logic are categorized as arithmetic instructions. When the result of the arithmetic operation is determined and is about to be stored in memory or a register, a condition code is set indicating whether the result is positive, negative, or zero. Compare instructions set the condition bits based on a greater than, less than, or equal to comparison of two registers or a register and the contents of a memory address.

Two other bits in SR1 are set as a result of computational or shifting instructions. The overflow designator is set whenever an arithmetic or shift operation requires more bits than are provided for in a register (16 bits).

The carry designator is set when an arithmetic operator generates a carry beyond the most significant bit of the register.

The AN/UYK-20 allows five different types of operand formats: single-length, byte, literal, optional floating point, and double-length. Single-length operands are 16-bit values with the sign bit assumed to be in the most significant bit. In arithmetic calculations, the single-length word is assumed to be a two's complement integer. Byte operands are considered 8-bit unsigned integers, and can be the most significant or least significant half-word of a memory location. Literal operands are 4-bit unsigned integers denoted by the 'm' field of a literal type instruction. Floating point operands are formed using two adjacent registers or memory locations with fields containing the sign of the fraction, the characteristic, and 24 bits of the fraction.

Double-length operands are concatenated from two adjacent registers or two adjacent memory addresses. The most significant half of the double-length operand is contained in the low-order register or memory address, and the least significant half in the next sequential register or address. The sign bit for both words resides in the high-order bit position of the most significant half-word and when used in an arithmetic calculation, the double-length operand is treated as a two's complement integer. Instructions involving double-length operands require the low-order

register or memory address to be even, since the adjacent cell address is formed by 'OR'-ing a 1 in the least significant bit of the address.

2. Instruction Format

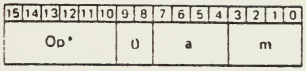
The AN/UYK-20 has five different instruction word formats, designated by two-letter mnemonic codes. These codes are: RR (Register-Register), RL (Register-Literal), RI (Register-Immediate), RK (Register-Constant), and RX (Register-Index). Each of these formats are designated in the instruction word by a combination of the opcode field and the subfunction code. Registers are identified in the 'a' and 'm' fields of the instruction, and are referred to by the notation Ra and Rm. The 'v' field is treated as an address or arithmetic constant, depending on the instruction (Figure III-5).

The format RR single-word instructions perform operations involving the registers specified by the 'a' and 'm' fields. No memory references are made to access operands, causing considerable savings in execution time. The 'a' or 'm' field may be used to index special operations on registers.

Format RL instructions use a 16-bit instruction word, using one or two general registers. The 'a' designator selects the general register Ra or register pair Ra, Ra + 1. The 'm' designator contains the 4-bit literal value which will be used in the instruction.

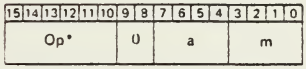
INSTRUCTION FORMATS

RR



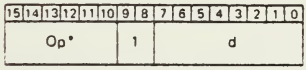
a selects R_a; m selects R_m.

RL



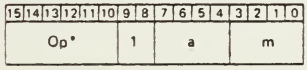
a selects R_a; m contains 4-bit constants.

RI-1
Local
Jump



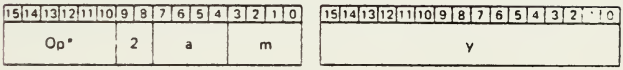
d is signed number of instructions to jump, relative to current position.
(+ is forward; − is backward)

RI-2
Indexed
Memory



a selects R_a; m selects R_m. R_m selects memory address.

RK



a selects R_a; m ≠ 0 selects R_m, m = 0 selects 0.
Operand = y + R_m or 0.

RX

a selects R_a, m ≠ 0 selects R_m, m = 0 selects 0.
y + R_m or 0 selects memory address.

*Op is operation code

Figure III-5 [23]

RI format single-word instructions are separated into two catagories. RI Type 1 instructions are local jump instructions where the P-register is increased or decreased by the 'd' field in the instruction. The 'd' field represents the two's complement deviation value and allows the P-register to be altered a maximum of +177 octal or -200 octal. RI Type 2 instructions involve operations that use the general registers Ra and Rm, and a memory address specified by the contents of Rm.

The format RK instructions contain 32 bits of information. The upper 16-bits contain the operation code and the designator fields. The 'a' field selects the general register Ra, and the 'm' field denotes how the next word, containing 'y', is to be used. If 'm' equals zero, then the effective operand address Y, equals 'y'. If 'm' does not equal zero, then Y is formed by adding the contents of Rm to 'y'.

Format RX are double-word instructions similar to the RK instruction that use direct and indirect addressing techniques determined by the 'm' field. If the 'm' field equals zero, then direct addressing without indexing is employed, with the effective operand address formed from the 'y' field itself. If the 'm' field equals 1 through 7, 11, 13, 15, or 17 (octal), then direct addressing with indexing is employed. The effective operand address is formed by adding the contents of Rm to 'y'. An 'm' field value of 10, 12, 14, or 16 (octal) indicates indirection is to be

employed, and the indirect address control fields in status register 2 contain information which is used to generate the effective operand address or a pair of indirect words. When the control fields equal 0 or 1, then direct addressing with indexing results. If the control field equals 2, then the contents of the first indirect word (IW1) is located at 'y'. Finally, if the control field equals 3, then IW1 is located at 'y' indexed by the contents of Rm. Indirect word format is shown in Figure III-6. Cascaded indirection is possible provided that the indirect words are properly encoded.

Byte addressing is accomplished using RX format instructions. A byte identifier (B) is used to determine which half-word (8-bit) is to be referenced. If B equals 0, then the most significant half-word in address Y is the operand byte. If B equals 1, then the least significant byte at Y is the operand. The least significant bit in the indexing register is used as the byte identifier, and the remaining 15 bits are used as the indexing value for finding Y. Indirect byte operand addressing formulae are included in Figure III-6. The following formulae apply for direct byte operand addressing:

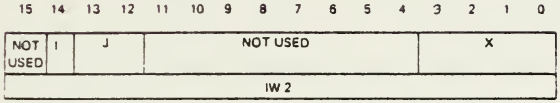
m=0, Y=y, and B=0

m=1-7, 11, 13, 15, or 17 octal

Y=y + (Rm)/2 and B=LSB of (Rm)

The preceding section has described the fundamentals of the AN/UYK-20 instruction formats. References 21 and 23 should be consulted for further information concerning instruction formats and decoding.

INDIRECT WORD FORMATS



OCTAL J VALUE	ADDRESS DETERMINATION
0	Y = IW2; if byte mode, upper byte is used.
1	Y = IW2 + (Rx); if byte mode, LSB of Rx determines byte. *
2	Y = IW2 + (Rm); if byte mode, LSB of Rm determines byte. *
3	Y = IW2 + (Rm+1); if byte mode, LSB of Rm+1 determines byte. *

I = 0, DIRECT ADDRESSING, OPERAND AT ADDRESS Y

I = 1, CASCADED INDIRECT ADDRESSING, NEW INDIRECT WORD 1 AT ADDRESS Y

* To determine the operand address when in byte mode, the contents of R_x, or R_m, or R_{m+1} are right shifted 1 bit position and zero filled in the left most position

Figure III-6 [23]

IV. BURROUGHS D-MACHINE

A. HARDWARE DESCRIPTION

The microprogramming facility at the Naval Postgraduate School is composed of a Burroughs Interpreter-Based system. This system possesses the characteristic of being dynamically microprogrammable and is designed using a simple building block structure. A typical system is made up of a number of interpreters (processors), main memory modules, and input/output devices, along with a switch interlock device (SWI) controlling data flow on the data bus connecting the interpreters to main memory and peripheral devices. The heart of the system is the interpreter, also referred to as the D-machine.

A D-machine possesses five functional modules: the logic unit (LU), the control unit (CU), the memory unit (MU), nanomemory (NM), and microprogram memory (MPM). The system presently installed in the Computer Science Department combines nanomemory and microprogram memory into one functional unit. The architecture of the D-machine is designed around 8-bit word slices. Word lengths from 8 bits to 64 bits are permissible using the same functional unit (Figure IV-1).

INTERPRETER BLOCK DIAGRAM

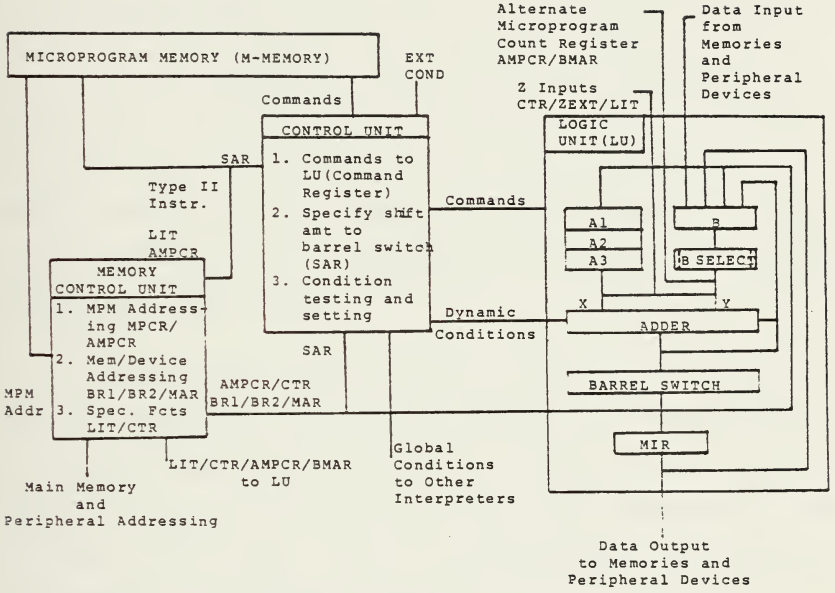


Figure IV-1 [o]

The D-machine used for this thesis has been configured as a 32-bit word processor. Reference 17 provides a thorough and concise description of the architecture of an interpreter-based microprogramming system. Reference 6 details the specifics of D-machine microprogramming which must be thoroughly understood by the programmer.

1. Logic Unit

The D-machine's logic unit performs shifting, arithmetic, and logic functions and contains scratch pad registers and data interfaces for the switch interlock. All adder operations are performed using two's complement arithmetic, and shifting is accomplished in a matrix of gates called the barrel switch.

The scratch pad registers A1, A2, and A3 are identical in function. They act as temporary storage registers within the D-machine and serve as primary inputs to the adder. The control unit determines which register will be an input to the adder. Any of the A-registers can receive the output from the barrel switch.

The B register functions as the primary external input interface from the switch interlock. It acts as the second input to the adder and can receive certain direct inputs from the adder's arithmetic operations. The b register can be loaded from any of the following sources:

a. The barrel switch output.

b. The adder output.

c. External data from the switch interlock (or control panel switches).

d. The memory information register (MIR).

e. The complements of four bit or eight bit carries.

f. The barrel switch ORed with the adder, external data from the switch interlock, or MIR.

The output of the B register is filtered prior to gating into the adder. The 'filter' consists of true/complement selection gates which are divided into three sections: the most significant bit, the least significant bit, and all the remaining central bits. The output of this filter is the independent result of these sections and may be either all zeroes, all ones, or the true contents or one's complement of the contents of the respective bits of the B register.

The memory information register is the interface to the switch interlock. MIR functions as a buffer to main memory or to a peripheral device. It is an output destination of the barrel switch and its output can be sent to the B-register or the switch interlock.

The adder in the logic unit performs all arithmetic functions and can be categorized as a version of a carry look-ahead adder. The control unit can gate various combi-

nations of A, B, and Z inputs to the adder. An 'A' input is defined as an input from one of the three scratch pad registers. A 'B' input is the output of the B register's true/complement filter gates. A 'Z' input is an external input to the logic unit and can originate from one of the following sources:

a. The output of the counter in the memory control unit (MCU) which is gated to the most significant eight bits of the adder with the remaining bits zeroed.

b. The output of the literal register in the MCU which is gated to the least significant eight bits of the adder with the remaining bits zeroed.

c. An optional input which is gated into the middle bytes of the adder with the most and least significant bytes zeroed.

d. The output of the alternate microprogram count register (AMPCR) in the MCU which is gated into the least significant 12 bits of the adder (13 bits for 8K micromemory machines) with all other bits being zeroed.

e. All zeroes.

Two inputs, selected from the A, B, or Z sources, are always gated to the adder. These inputs are referred to as X-select and Y-select. An X-select input may be either an A input or a Z input. If it is not specified, it is

assumed to be zero. A valid Y-select has either a B, Z or 1 as its input. Some Z inputs, however, are valid only as Y-select inputs. Any combination of valid X-selects and Y-selects are permissible addends, with an option of adding a one to the least significant bit. For subtraction operations, the value to be subtracted must be a Y-select; the Y-select input is subsequently two's complemented and gated to the adder. All binary Boolean operations between two adder inputs are allowed, and dynamic condition bits for overflow (AOV), all bits true (ABT), and most/least significant bit test (MST/LST) are available to the control unit for testing. Bit testing is a valuable feature for decoding instruction words.

The barrel switch is a matrix of gates that accepts parallel data from the adder and shifts the data any number of places to the left or right with zero fill. It also can right shift the word in an end-around fashion. The output of the barrel switch may be directed to any of the following destinations simultaneously:

a. The A registers.

b. The B register.

c. The memory information register.

d. The least significant 16 bits to the MCU registers.

e. The least significant three to six bits to the control unit shift amount register (bit length depending on the word size of machine).

2. The Control Unit

The control unit has five major sections: the shift amount register (SAR), the condition register (COND), part of the control register (CR), the decoder for microprogram memory content, and clock control. This module of the D-machine manages the functions of the processor, and is directly responsible for logic unit operation.

SAR and its associated logic control shifting operations by loading shift amounts into SAR and generating the required controls indicated by the current microinstruction for the barrel switch. In addition, the SAR's logic generates the 'word length complement' of the SAR contents where the complement is defined to be that amount which would restore the bits of a word to their original position after an end-around shift of N followed by an end-around shift of the 'complement' of N. For example, if an end-around right shift of 20 was required in a 32-bit D-machine, another end-around shift of the complement of 20 (12) would be required to restore the contents to its original value.

The condition register has four major functions:

a. It stores 12 resettable bits which are used as error indicators, interrupts, status, and lockout indicators.

b. It selects one of 16 condition bits for performing conditional operations. These 16 bits are composed of the 12 condition bits of the

condition register plus the 4 dynamic conditions generated by the LU adder during the present clock time.

c. It decodes bits from the memory for resetting, setting, or requesting the setting of designated bits of the condition register.

d. It resolves priority between interpreters in the setting of global condition bits (GC), thereby providing a method of controlling inter-interpreter lockout.

The control register stores the control bits of the 56-bit microinstruction that are not being used in the first phase of the execution cycle. The control register is subdivided into sections which are used by the memory control unit, the logic unit, and the control unit during the execution phase of a microinstruction. For a description of timing and phases, see section C of this chapter.

3. Memory Control Unit

The memory control unit provides the basic addressing interface between the D-machine and both main memory (S-memory) and microprogram memory (M-memory). One MCU can address 64K words (256K bytes) of main memory, and if the D-machine is configured with a second MCU, a maximum of 128K words can be addressed.

The memory control unit has three major sections:

a. The microprogram address section controls the addressing of microprogram memory and the sequencing of microinstructions. It contains the microprogram count register (MPCR), the alternate microprogram count register (AMPCR), the incrementer, the microprogram address controls register, and their associated logic. For standard 4K M-memories, the MPCR and AMPCR are 12 bits long. For 8K M-memories, MPCR and AMPCR are 13 bits in length (the D-machines installed at the Postgraduate School employ 8K M-memories). AMPCR is a Y-select input to the logic unit adder.

b. The memory/device address section contains the 8-bit memory address register (MAR), two 16-bit base registers (BR1 and BR2), output selection gates, and associated control logic. When forming a memory address, the lower eight bits of a base register and MAR are concatenated. The concatenated 24-bit contents of BR1/ BR2 and MAR (BMAR) is a valid Y-select input to the logic unit adder.

c. The Z register section contains two registers which are Z inputs to the logic unit adder. The literal register (LIT) is an 8-bit register into which constants are loaded. An 8-bit counter (CTR) is used in conjunction with a counter overflow condition bit to control iterative

looping. The Z register section also contains selection gates for the loadable counter and its associated logic.

4. Microprogram memory (M-memory)

A D-machine may have a dual or single microprogram memory scheme. As indicated earlier, the D-machines used in this emulation project had a single microprogram memory, consolidating the microprogram memory and nanomemory into one 56-bit programmable store memory, often referred to as M-memory. Microprograms, consisting of 56-bit microinstructions, are dynamically changeable by the user, thus distinguishing the D-machine as an extremely flexible computing device.

The sequencing of microprogram instructions is controlled by a condition bit procedure which determines the successor command to be executed. M-memory provides data to the condition testing logic which then determines which condition is to be tested. The output of the condition testing logic is a true/false signal that is gated to the successor selection logic. This logic then selects between the three true and three false successor bits also provided by the M-memory word. The three selected bits provide eight possible successor combinations:

a. WAIT Repeat the current instruction.

b. STEP Step to the next instruction.

c. SKIP Skip the next instruction.

d. JUMP Jump to another M-memory address.

e. RETN Return from a microprogram subroutine.

f. CALL Call a microprogram subroutine.

g. SAVE Step and save the instruction address.

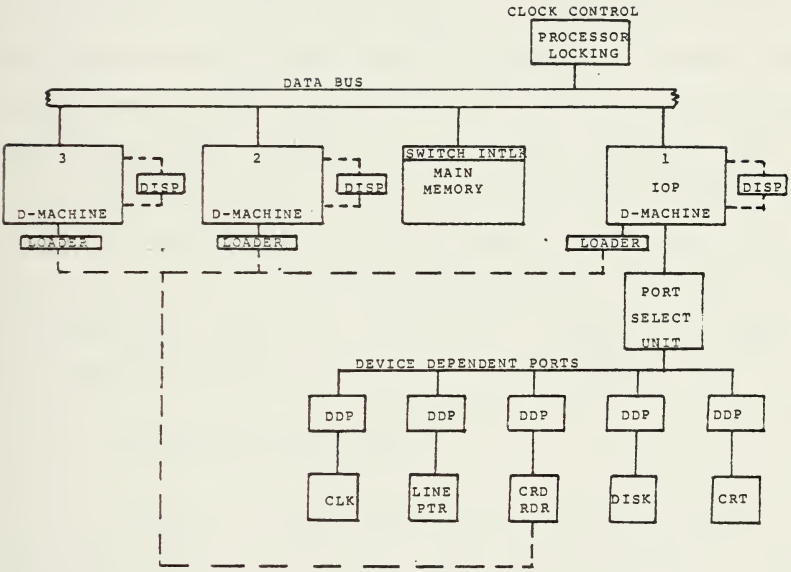
h. EXEC Execute one instruction out of sequence.

The particular successor command chosen controls gates which select the appropriate M-address from MPCR or AMPCR and provides incrementing logic for generating the next M-memory address. Except for the EXEC command, the MPCR is loaded with this M-memory address.

B. NPS MICROPROGRAMMING FACILITY

1. Physical Description

The Computer Science Department of the Naval Postgraduate School possesses a Burroughs Interpreter-Based System consisting of three interpreters (processors) also known as D-machines, a 64K 32-bit word, main memory module, a card reader, a dual cartridge disk drive, a line printer, and a Datamedia 2500 CRT functioning as a supervisor's console (Figure IV-2). All input and output is performed through a single D-machine processor, hardware configured with device dependent ports (DDP) for peripherals and the external clock.



NAVAL POSTGRADUATE SCHOOL MICROPROGRAMMING FACILITY

BURROUGHS INTERPRETER-BASED SYSTEM

Figure IV-2

After initial light-off and bootstrap, the system is configured into two Burroughs 6700 - LIFO ALGOL Stack-machines, each addressing 32K of memory and each communicating with the input output processor (IOP) in a pseudo-multiprocessing environment. Software, written in ALGOL, is provided which runs on the B-6700 system. A resident monitor control program and disk file manager control the maintenance of system files and the execution of jobs in a batch environment. Both D-machines compete for system jobs input from the card reader or CRT. Other software available includes an ALGOL compiler (a derivative of ALGOL 60), a microprogram translator called TRANSLANG, a line editor, and a simulation program for microprograms. TRANSLANG provides the medium by which microprograms are written. User microprograms loaded into a D-machine change its identity and destroy the Stack-machine previously loaded.

2. Input/Output Interface

Pivotal to the operation of the Burroughs system is the input/output interface. Only one processor, the IOP, communicates with peripherals, and the other D-machines, configured as Stack-machines, must compete for its services. The IOP communicates asynchronously, using a conventional 'handshake' method. Since all interpreters have access to the main memory module, a communications link has been established using the upper two 32-bit words of main memory. If a Stack-machine wishes to communicate with the IOP, it places a message into address 65,535 known as the 'mailbox',

and issues an interrupt (INT) to the IOP. The IOP periodically tests INT, and if set, will retrieve the contents of the mailbox (and mailbox + 1 if required) and perform the desired operation. The other Stack-machine is locked out from interrogating the IOP until it has completed processing the request. Normally, the operation requires transferring a buffer to some output device. When the IOP has completed honoring the request, it places a completion code in the mailbox and sets INT for the Stack-machine requesting the I/O. This interpreter must halt execution and check mailbox to see if the I/O was performed successfully, completing the handshaking process. This protocol permits both Stack-machines to perform input/output independently of each other, provided both maintain strict memory boundaries.

Another function of the IOP is interfacing character code formats between the peripherals and the other two D-machines. When the machines are configured as Stack-machines, characters are passed to the IOP in 6-bit BCL (Burroughs Common Language). The IOP must convert this character set to ASCII for output to the line printer and CRT. A similar translation must be made for input data converting from either EBCDIC or ASCII depending on whether the input source is the card reader or the CRT.

3. Memory Interface

Since all three D-machines must share the 64K of main memory, a priority scheme was developed to resolve

memory reference conflicts. The main memory module is actually a single-ported, 32-bit word, core memory which can be made to appear multiported using a switch interlock unit (SWI) developed by Burroughs. The switch interlock controls the main data bus of the system, and resolves conflicts using a priority scheme. The D-machine with the highest priority is the IOP, with the priority of the other two machines being relative to their physical proximity to the IOP. Once a memory reference has been made, a D-machine may continue execution without waiting for a completion signal from the switch interlock. Although this technique of memory referencing minimizes unnecessary delay, it restricts the program from changing the read or write addresses or the content of MIR (write only) prior to a completion signal.

C. MICROINSTRUCTION TIMING

The Burroughs D-machine initiates a microinstruction once every clock cycle. The D-machines utilized for the AN/UYK-20 emulation operated from a one mHz internal clock, which produced a clock pulse once every microsecond. A D-machine designed with an eight mHz clock, emitter-coupled logic (ECL), and a faster memory cycle time, however, could execute eight times faster. This implies that advances in circuit technology can permit emulations to achieve improved speed and performance with no change in the microprograms.

Every microinstruction is executed using one or more sequential time periods, called phase 1, phase 2, and phase

3. A phase is a constant interval of time equivalent to one clock duration measured from the trailing edge of each successive clock pulse. Some microinstructions only require phase 1 to complete execution. Some require phase 1 and phase 3, and still others require phases 1,2, and 3. A new microinstruction is initiated at each clock cycle, allowing for overlapping of microinstruction execution in phase 1 and phase 3.

Microinstructions consist of two types. In a type 1 microinstruction, events can take place in all three phases:

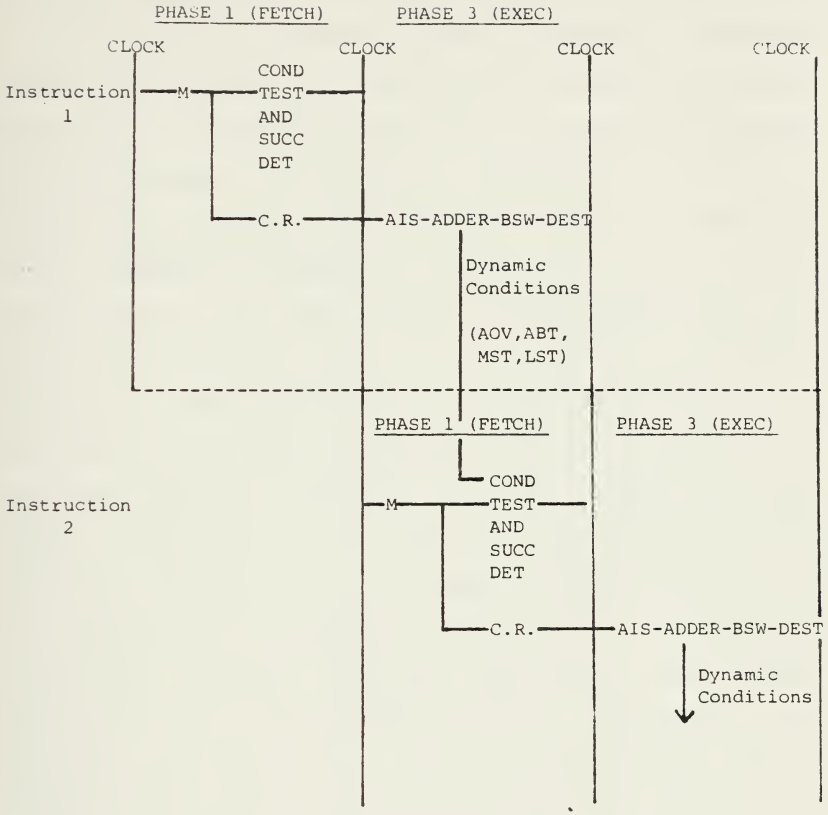
Phase 1: condition testing, (conditional) external operations or (conditional) logic operation initiation after completion of a prior logic operation, and successor M-memory address control.

Phase 2: a holding phase for phase 3 logic operation controls.

Phase 3: the completion phase for logic unit operations and destination register gating specified by logic operation.

During the optional phase 2 period, a type 1 microinstruction execution completion is held in abeyance while a subsequent type 2 instruction is executed. A type 2 microinstruction requires only phase 1 to complete execution, and involves literal assignments to three registers: LIT, SAR, and AMPCR. Phase 2 may also be initiated if the

next sequential type 1 instruction does not execute its conditional logic operation and therefore can complete its execution in phase 1 (Figure IV-3). Appendix D of Ref. 6 has a complete discussion of microinstruction timing.



M - MPM ACCESS TIME
COND TEST AND SUCC DET - CONDITION TEST AND SUCCESSOR DETERMINATION
BSW - BARREL SWITCH
DEST - BARREL SWITCH OUTPUT DESTINATIONS; I.E., REGISTERS (B,CTR,ETC.) AND THEIR INPUT LOGIC
C.R. - COMMAND REGISTER AND ASSOCIATED LOGIC
AIS - ADDER INPUT SELECTION FROM COMMAND REGISTER

Timing Analysis, Type I Instructions

Figure IV-3 [17]

D. TRANSLANG

Microprogramming on the D-machine is accomplished using a microtranslator/assembler called TRANSLANG. TRANSLANG allows the programmer to write microinstructions mnemonically without concentrating on the bit patterns that compose the microinstructions themselves. TRANSLANG is written in ALGOL, the language of the B6700 Stack-machine. Nearly the entire language of TRANSLANG is composed of reserved words recognized by the ALGOL program. Each reserved word has a special meaning which causes the translator to construct particular microinstructions. A TRANSLANG instruction is equivalent to one microinstruction consisting of the set of parallel D-machine functions performed during the clock phases. TRANSLANG is a free form language and instructions may be written in almost any order. Multiple instructions may appear on a line, separated by a period '.'. TRANSLANG constructs include iterative mechanisms, input/output, assignment functions, control transfers, and Boolean, and computational operations. In addition, TRANSLANG permits label definitions and symbolic references for program control flow. Reference 6 is the programming manual for TRANSLANG and contains the complete syntax for the language. Appendix A of Ref. 10 documents additions to the TRANSLANG instruction repertiore.

A microprogrammer may construct complicated microinstructions that perform many different tasks, some interacting closely with D-machine clock timing. Microinstruction

gating to several devices permits a single TRANSLANG instruction to accomplish some or all of the following actions:

a. test a condition.

b. set/reset a condition.

c. initiate an external operation.

d. add.

e. shift the result of an add.

f. store the result into several registers.

g. increment a counter.

h. complement a shift amount.

i. determine the successor microinstruction.

By judiciously composing his microprogram, a programmer may minimize execution time by taking advantage of microinstruction phase overlap and using highly parallel microcode.

The TRANSLANG assembler constructs an object program consisting of non-relocatable 56-bit microinstructions. TRANSLANG maintains a cross reference table that resolves label references during assembly. The object code created is stored on a disk and may be loaded into the micromemory of a D-machine using a special control word recognized by the operating system. Once loaded, the D-machine assumes the control structure dictated by the users microprogram.

V. AN/UYK-20 EMULATOR

A. EMULATION DESIGN

1. Functional Components

The architecture of the AN/UYK-20 emulator microprogram was developed using general guidelines provided by references and previous emulation experience on Burroughs D-machines [10]. The first decision incorporated in the emulator design was to integrate the entire emulation within one D-machine. Since the D-machines had the capacity to handle nearly 8K of microinstructions, no microprogramming capacity limitations were envisioned. The design objectives of a modularized, well-documented, structured microprogram could also be realized.

With the emulator entirely contained in one D-machine, several secondary benefits also existed. First, the emulator was more immune to hardware problems. If one D-machine was malfunctioning, the emulator could still be run on the alternate D-machine. Second, it was possible to have two AN/UYK-20 emulators resident in the system at one time. Not only would two emulators speed up testing of the individual emulation, but they would also permit their eventual use in a system configured for AN/UYK-20 multiprocessing. The third and final benefit of a totally integrated

emulator was recognized during the design of the input/output controller (IOC). Since the AN/UYK-20's IOC was capable of independent processing, an emulation of its IOC would not be possible within the same D-machine. An emulation of the IOC could be accomplished in a second D-machine, which would behave as an independent channel for the D-machine configured as the AN/UYK-20 processor. Although this emulation does not attempt to emulate the AN/UYK-20 IOC, the fact that a second D-machine exists makes its implementation a realistic extension to the project.

The emulator program organization was created following the basic guidelines of Ref. 17. The loader occupied the lowest section of M-memory with the emulator microcode following sequentially. Microprogram control passed from the loader to the emulator via the execution of a 'G' card which signified execution commencement.

The emulator microprogram was organized into three modules positioned such that forward address referencing would be minimized in the TRANSLANG assembler to save time and space. The utilities section was in the lowest portion of the emulator, since its routines were referenced frequently by the succeeding sections. Individual subroutines within the utility section were organized alphabetically.

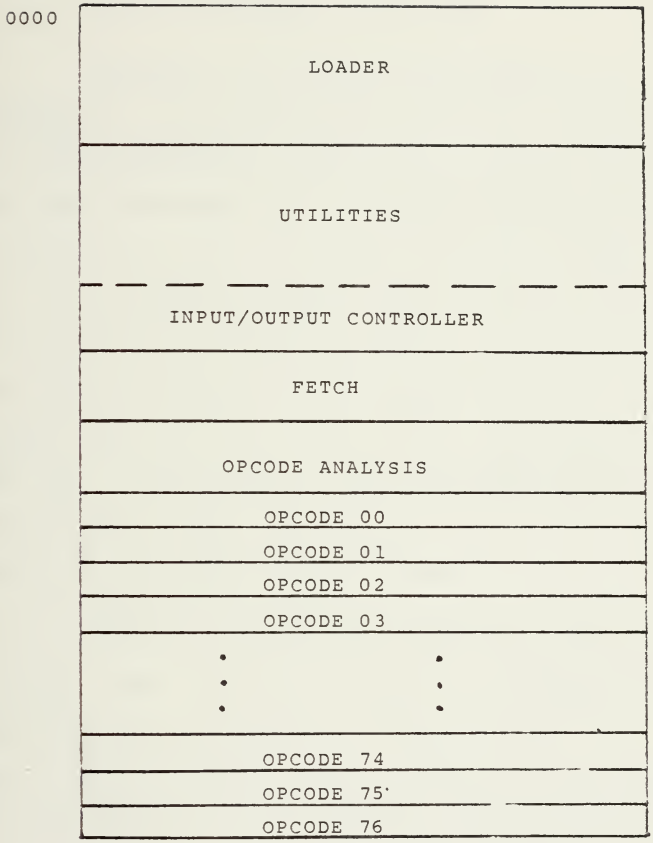
The instruction and memory fetch routines comprised the next module. These routines incorporated all fetch options available in the AN/UYK-20 emulator.

The opcode routines occupied the last section of the emulator. The opcodes were arranged numerically with further indexing determined by the remaining fields of the individual instruction. Figure V-1 shows the M-memory mapping of the AN/UYK-20 emulator layout. Appendix F contains the emulation program listing.

2. Main Memory Organization

Of primary importance in the emulation design was the logical organization of the Burroughs system's main memory. Since the AN/UYK-20 was a 16-bit word machine, it was decided that only the lower half-word of a Burroughs 32-bit word would be used by the emulator. This design restriction permitted the addressing structure of the AN/UYK-20 to be directly projected on the D-machine. A one-to-one correspondence between the memory address of the AN/UYK-20 and the Burroughs system was also achieved.

Since the number of high-speed registers available in the D-machine was small, a 1K portion of main memory was reserved for the AN/UYK-20 addressable register set, the page address registers, the temporary storage space, and the emulation buffers. The mapping of the AN/UYK-20 high-speed registers to main memory greatly simplified the microprogramming requirements of the emulation, but added considerable execution time overhead. Memory access times for the mapped registers were much slower than the actual transfer



AN/UYK-20 EMULATION ORGANIZATION

Figure V-1

rates of the AN/UYK-20 registers. Figure V-2 shows the complete main memory mapping of the emulation reserved storage area.

3. Emulation Program Status Word

Although the emulation duplicated all the AN/UYK-20 registers, the registers which comprised the program status word (PSW) possessed unique characteristics. Status register 1 was combined with the program address register to form a single 32-bit PSW. The emulator's PSW always co-existed in the A1 register of the D-machine and in main memory during execution of AN/UYK-20 programs. The fields of SR1 were modified to include certain emulator toggles, along with the normal condition bits (Figure V-3). The condition bits for DMA and non-destructive read only (NDRO) mode were removed from the SR1 since they were hardware features of the AN/UYK-20 that would not be emulated.

Status register 2, the remaining 16 bits of the AN-UYK-20's PSW, was not resident in any D-machine registers during emulation execution for two reasons: the emulation could not afford the luxury of 48 bits of reserved register space, and SR2 was less frequently referenced than SR1 and the P-register. Consequently, SR2 had to be read from its reserved location in main memory. The contents of the upper 16 bits of SR2's memory location also contained additional emulation toggles which were used by the debugger package (Figure V-3).

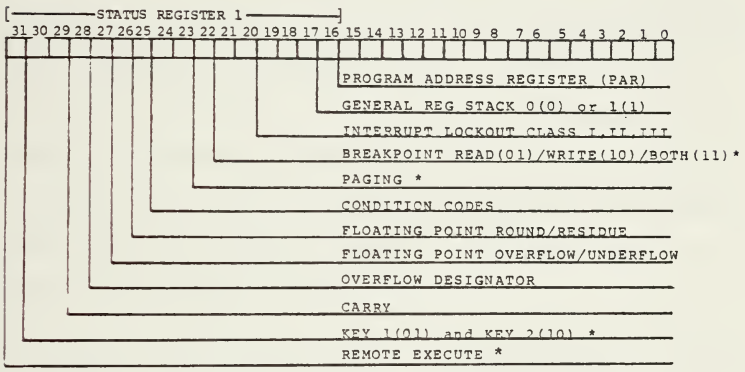
MAIN MEMORY MAPPINGS

DECIMAL ADDRESS	OCTAL ADDRESS	USE
0 - 15	0 - 17	GENERAL REGISTER STACK 1
16 - 31	20 - 37	GENERAL REGISTER STACK 2
32	40	PROGRAM STATUS WORD
33	41	BREAKPOINT REGISTER
34	42	STATUS REGISTER 2 (SR2)
35	43	NEXT LOAD ADDRESS
36	44	CLOCKTIME
37	45	REAL TIME CLOCK
38 - 43	46 - 53	WORKSPACE (TEMP STORAGE)
44 - 49	54 - 61	STACK (TEMP STORAGE STACK)
50 - 53	62 - 65	I/O COMMAND WORDS (IOCW)
54 - 119	66 - 167	UNUSED
120 - 121	170 - 171	HEX ADDRESS FOR INPUT
122 - 141	172 - 215	INPUT CARD BUFFER
142 - 152	216 - 230	UNUSED
153 - 185	231 - 271	OUTPUT PRINT BUFFER
186 - 205	272 - 315	CRT BUFFER
206 - 229	316 - 345	ERRORLIST
230 - 767	346 - 1377	UNUSED
768 - 1023	1400 - 1777	PAGE ADDRESS REGISTERS

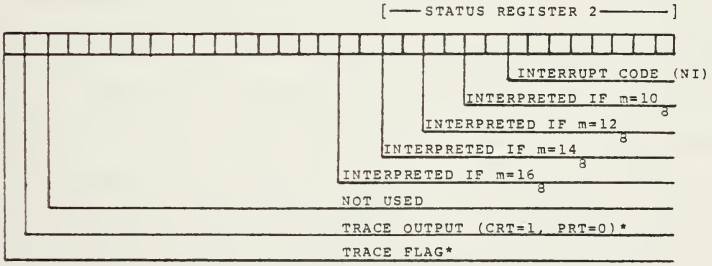
Figure V-2

AN/UYK-20 EMULATION
PROGRAM STATUS WORDS (PSW)

ACTIVE PSW: RESIDES IN LU REGISTER A1 (MEMORY ADDRESS 32)



INACTIVE PSW: RESIDES IN MEMORY ADDRESS 33



* FIELDS ADDED FOR USE BY THE EMULATOR

Figure V-3

4. D-Machine Registers

With only a limited number of high-speed registers available in the D-machine, the emulation design had to include a well-developed plan for their usage. Since only seven registers in the D-machine were 16 bits or longer, they were used exclusively for manipulating the AN/UYK-20 addresses, instructions, and data. The register environment had to be consistent throughout every execution cycle to allow utility routines to be called in the same manner by all opcode subroutines. The primary goal of the emulation was to use as few memory references as possible to conserve execution time and memory space. Judicious use of the existing registers was a necessity.

Several factors had to be considered before selecting a register for a particular function. The most important consideration was its flexibility within the D-machine. The B register, for example, was used as a general purpose register, since its contents could be gated through a masking filter prior to being utilized. A second factor was the type of operand it could contain. Double-word operands (32-bit) could only be stored in the 32-bit logic unit registers while 16-bit operands could also be stored in the memory control unit registers.

The A1 register contained the emulation's 32-bit program status word (PSW) at the commencement of the emulator program. It was not affected by individual instruction

microcode, except when incrementing the PAR, or when a particular instruction modified the PSW. Emulator toggles resident in SR1 could not be altered by an AN/UYK-20 instruction because their settings were independent of program execution.

The A3 register held the instruction word for the duration of its execution cycle. Each field of the instruction could be decoded and interpreted from the A3 register without having to retrieve it from memory. Once the instruction had been completely decoded, A3 was made available as a scratch pad register.

The A2 and B registers were used as scratch pad registers during each opcode execution cycle. In general, their contents were volatile, except when they were specifically documented in the the program listing. The contents of A2, for example, was not altered in the EMULIN subroutine, because of its use in the calling fetch routine. A2 and B were manipulated as either single or double-word operands during the arithmetic operations.

The only remaining logic unit register was the memory information register (MIR). MIR was used for storing information into memory and as a temporary storage location. Intermediate results were deposited in MIR during instruction execution and returned through the B register.

The base registers, BR1 and BR2, were used as storage for addresses and single-length operands, and for

temporary storage of intermediate results. In addition, all memory addressing in the emulation was accomplished using the lower 8 bits of BR2 and MAR (MAR2). These memory control unit registers had to be used carefully, because they required a sequence of several microinstructions to properly reference their contents.

B. LOADER

The loader incorporated into the AN/UYK-20 emulation provided a simple mechanism for loading AN/UYK-20 instructions into main memory (S-memory) of the Burroughs microprogramming system (Figure V-4). Its control word repertoire was flexible, allowing a variety of AN/UYK-20 program environments. Job control statements were included to execute and halt individual programs anywhere in S-memory.

The loader module consisted essentially of a scanner and a translator written in the microcode. Information was read into a 20-word buffer from cards or CRT input, then the buffer contents were scanned for control code consisting of one or more characters. Once these characters were interpreted, control was passed to the translator section which decoded the rest of the data in the buffer and performed the required function. The translator section consisted of a variety of routines that handled specific control words in the loader repertoire. The loader control statements, however, had to appear in a logical sequence (See Appendix A). All loader control statements are contained in Appendix B.

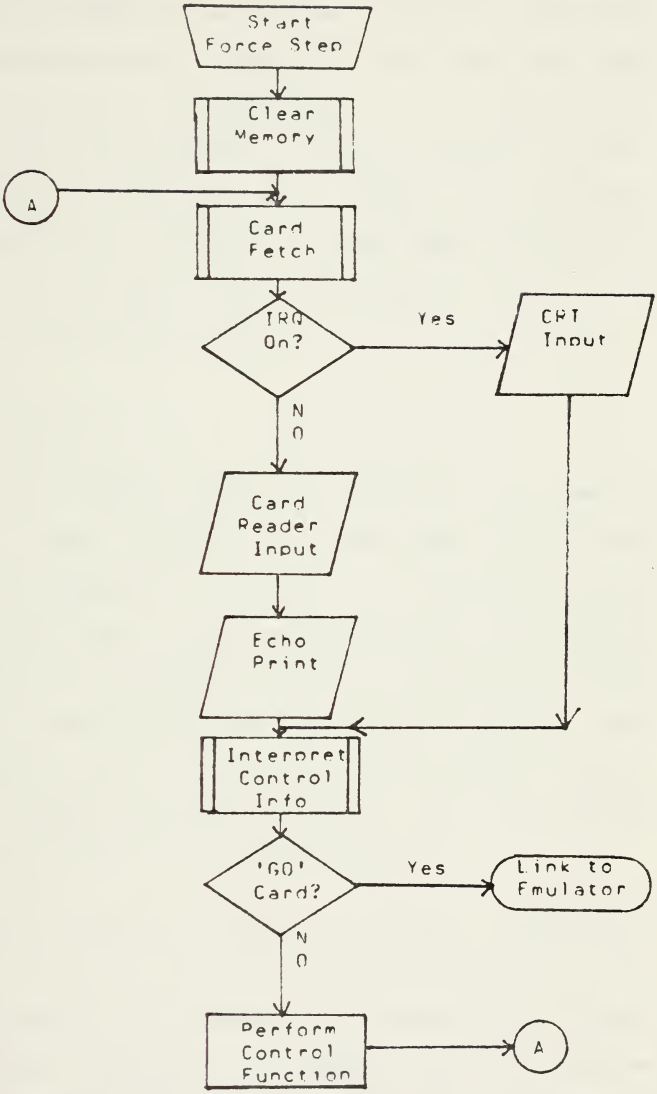

Start
Force Step
Clear
Memory
A
Card
Fetch
IRQ
On?
Yes
CRT
Input
N
O
Card
Reader
Input
Echo
Print
Interpret
Control
Info
'GO'
Card?
Yes
Link to
Emulator
N
O
Perform
Control
Function
A

Figure V-4

The actual interface pipeline between the loader and the emulator consisted of two small emulator utility routines which started and stopped the external clock of the Burroughs microprogramming system. These routines were inserted for emulation timing purposes, and provided a 'stop watch' for AN/UYK-20 programs. The time recorded (in milliseconds) by this 'stop watch' was placed in a reserved memory location in the emulation memory map, and could be read using either a trace control instruction ('T'), or a machine status control word ('M').

C. THE FETCH MODULE

In order to emulate AN/UYK-20 execution and memory referencing, fetch microcode was developed which incorporated memory addressing algorithms and instruction fetch routines. Since both data and instructions were equally accessible from the processor, the memory addressing scheme was closely linked to the instruction fetch concept. Data and instructions could be interspersed throughout memory, and proper program execution required that the program address register point to an instruction word.

The emulation used two routines for memory addressing, EMULIN for reading, and EMULOUT for writing (Figures V-5, V-6). These subroutines performed both paging and breakpoint checking, depending on toggles set in the program status word. Paging was incorporated into the emulation in order to gauge the execution overhead required in emulating

the AN/UYK-20's paging scheme. The paging scheme implemented in the emulator divided main memory into 256-word pages, instead of 1024-word pages used by the AN/UYK-20. Since the Burroughs D-machine was organized for 256-word pages, the microprogramming required for the paging was straightforward. The 256 page address registers resided in the emulator's memory mapping, each initialized to the page number corresponding to their relative address (0-255). The paging algorithm imitated the AN/UYK-20's method of page addressing, and included the setting of a page modification bit.

The breakpoint option was added to provide a method of debugging AN/UYK-20 programs, once the emulation was completed. EMULIN and EMULOUT tested toggles set in the PSW to determine if breakpoint read, write, or both was desired.

The memory addressing convention required all memory references from 1K to 64K to use EMULIN and EMULOUT. All memory references to the memory mapping area (0-1023) did not use these routines, but instead utilized absolute memory referencing microcode.

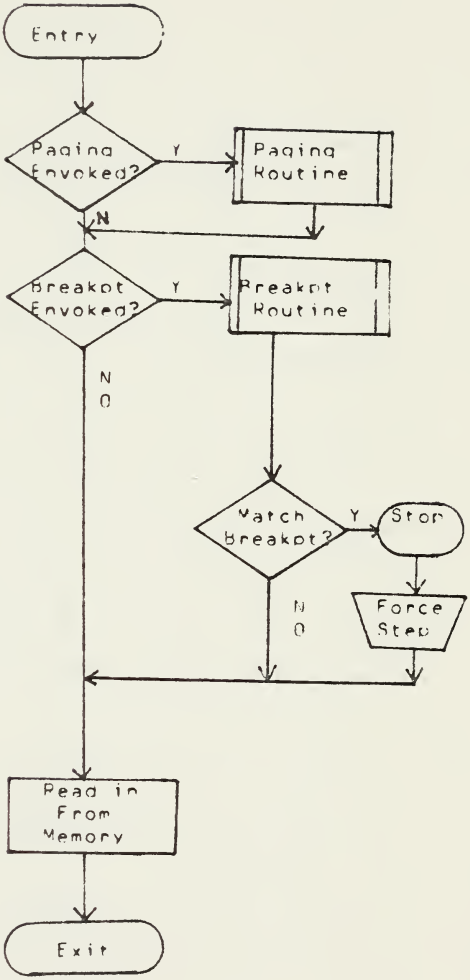

Entry
Paging Envoked?
Y
Paging Routine
N
Breakpt Envoked?
Y
Breakpt Routine
N
O
Watch Breakpt?
Y
Stop
N
O
Force Step
Read in From Memory
Exit

Figure V-5

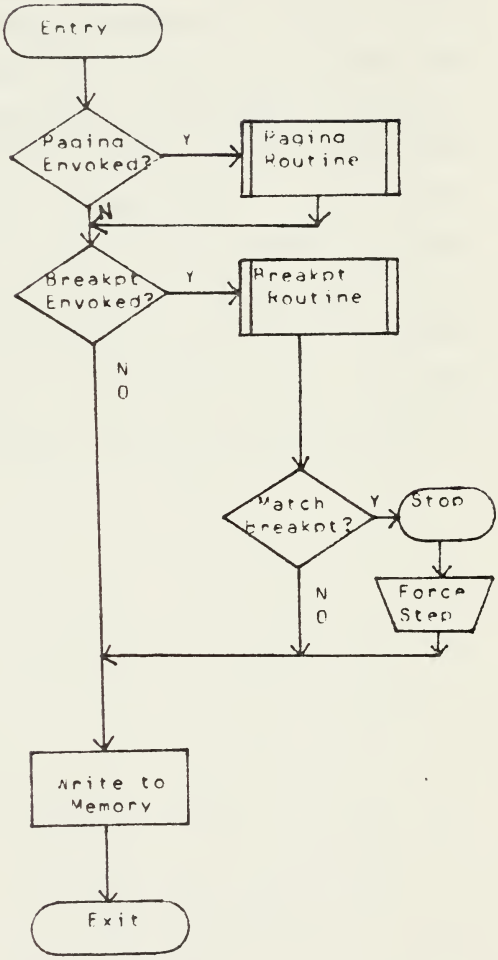

Entry
Pagina Envoked?
Y
Pagina Routine
N
Breakpt Envoked?
Y
Breakpt Routine
N O
Match Breakpt?
Y
Stop
N O
Force Step
Write to Memory
Exit

Figure V-6

The instruction fetch routine (IFETCH) retrieved instructions from main memory based on the contents of the program address register. Since IFETCH used EMULIN to read the instructions from memory, it could operate in a paging environment, with or without breakpoints (Figure V-7). IFETCH was also responsible for incrementing the PAR, and for testing the trace toggle prior to fetching an instruction. IFETCH was capable of retrieving any instruction word in the program address space (1024-65,535). In the event that the upper memory limit was reached, IFETCH would set the PAR to 1024 and continue execution. Trace toggle testing was inserted into IFETCH as part of the built-in debugging package. Since IFETCH was called prior to every instruction, it was the logical choice for placing a call to the debugger.

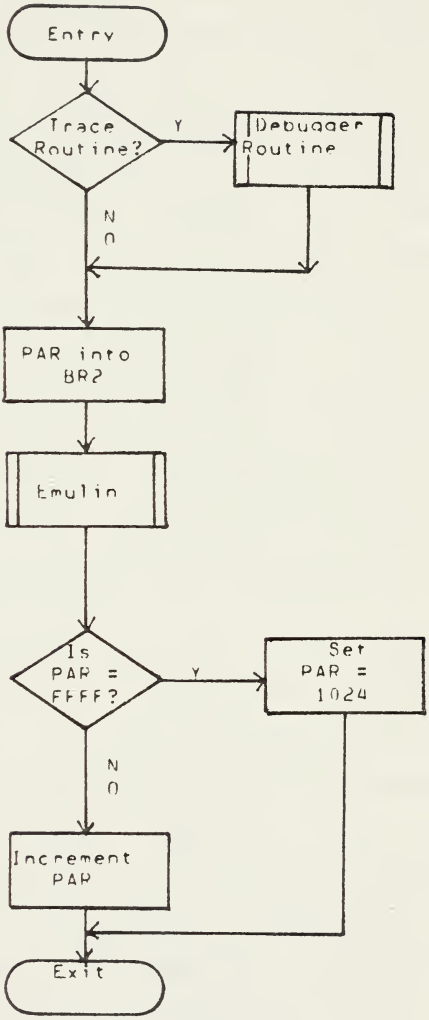

Entry
Trace Routine?
Y
Debugger Routine
N O
PAR into BR2
Emulin
Is PAR = FFFF?
Y
Set PAR = 1024
N O
Increment PAR
Exit

Figure V-7

D. OPCODE IMPLEMENTATION

All of the 205 individual instructions emulated were microprogrammed using an identical instruction decoding mechanism. The routines that performed the required operations were terminal nodes on a large tree network, whose root was the OPCODE routine. OPCODE fetched an instruction and isolated the operation code field. The binary value of the opcode field was an index to an operation jump table containing individual opcode M-memory addresses. Within each opcode routine was microcode which isolated the subfunction 'f' field of the instruction and used its value as an index to the next level of opcode analysis. Depending on the opcode, this last jump could identify which instruction was to be performed. If it did not, further analysis of the 'm' or 'a' fields provided the final index to the instruction. Instruction formats are described in Appendix C.

After the instruction had been executed, control passed back to the opcode routine, and then the execution cycle was repeated for the emulation of the next AN/UYK-20 instruction. The AN/UYK-20 programmer could cause this microprogram loop to be exited by either inserting an executive return instruction (03,0,a,00) which caused a 'priority interrupt' and halted program execution, or by coding an instruction that was not implemented, not assigned, or caused a division overflow. The last three cases caused an execution fault, while the former resulted in normal program termination (Figure V-8).

AN/UYK-20 EMULATION GENERAL STRUCTURE

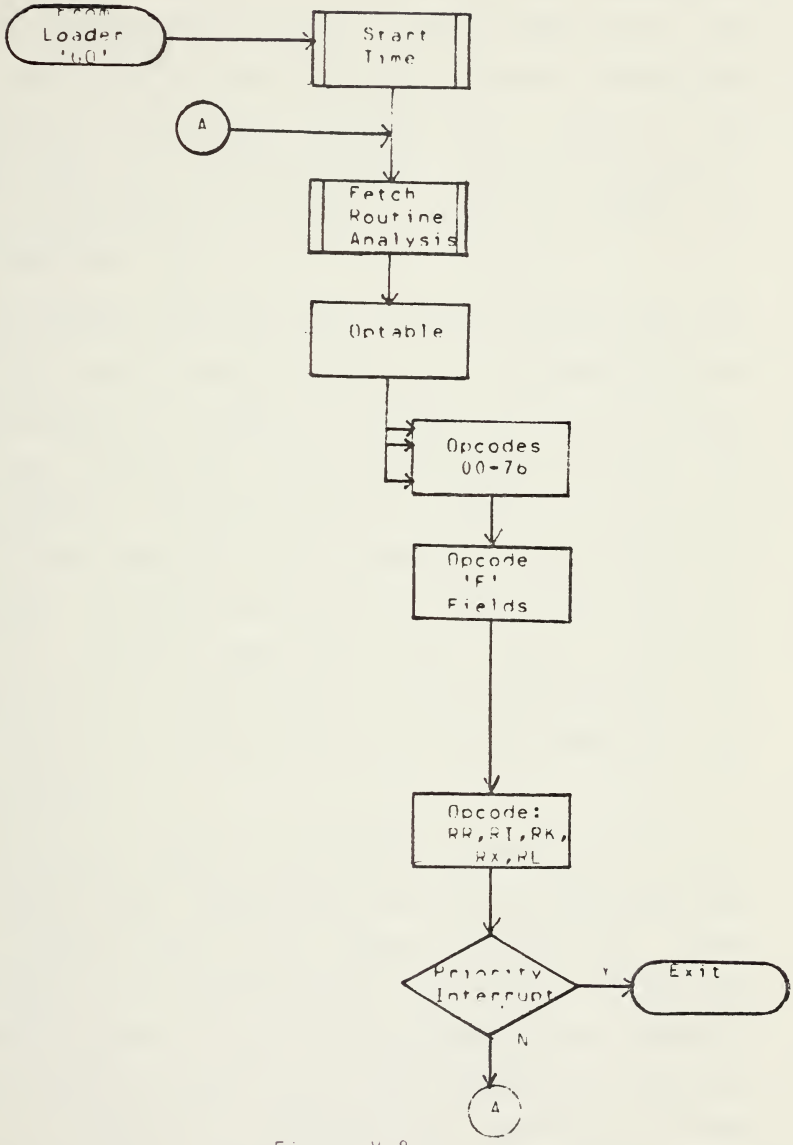


Figure V-8

The instruction fetch mode of OPCODE assumed that the PAR always pointed to an instruction word. Double-word instructions always had their 'y' field fetched after they were determined to be two words in length. Conditional double-word instructions performed their condition test before the 'y' field was fetched. If the test failed, the PAR was incremented by two prior to returning to OPCODE for continued execution.

The OPCODE routine was also written to accommodate the AN/UYK-20 'remote execute' instruction. When this operation was performed, a bit was set in the PSW which would indicate that one instruction out-of-line was being executed. For this operation, the current PSW was stored in memory, and the PAR was loaded with the address of the instruction to be executed. The OPCODE routine always checked the remote execute bit during an instruction cycle. If the bit was set, it would fetch the instruction indicated by the remote execute PAR, and restore the actual PAR, incremented by two, into the PSW.

Some of the opcode repertoire of the AN/UYK-20 was not emulated. Those instructions that were not emulated, however, retained slots in the opcode hierarchy for future inclusion. Any instruction not implemented by the emulator caused the machine to fault, and printed an error diagnostic on the selected output device ('NOT IMPLEMENTED - EXECUTION ENDS'). Similarly, locations were reserved for those instructions not assigned by the AN/UYK-20 were reserved

locations in the emulation . This permitted the emulator to be responsive to any future AN/UYK-20 hardware modifications. Whenever an instruction to be executed was not assigned, the emulator generated a fault interrupt and printed an error diagnostic on the selected output device ('FAULT INTERRUPT - EXECUTION ENDS').

E. UTILITIES

Although each emulated instruction performed different operations, each depended upon a common set of utility subroutines to accomplish their task. These subroutines varied in complexity, but each performed a function that contributed to successful instruction execution. A simple operation often required register addressing, condition code setting, and memory addressing, before the task was completed.

Utility subroutines were included in the emulation whenever feasible to simplify microprogramming and to alleviate programming redundancy. These subroutines were called using the successor command constructs available in TRANSLANG. Depending on the purpose of the utility routine, parameters were passed via D-machine register(s) or condition bit(s). This information was utilized by the utility in determining what operation was to be performed. In the carry subroutine, for example, a local condition bit was passed which indicated the appropriate condition code to be inserted into the PSW. A more complex example, the RX

format utility routine, required two parameters to be set by the instruction. Register A3 contained the instruction word and LC2 was set or cleared depending on whether or not byte formatting was required. The RX routine called other routines and could perform considerable processing before the final result, the effective operand address, was returned to the calling routine via the B register.

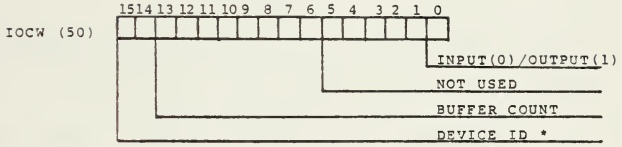
The utility section encompassed a number of routines which performed complex data manipulation. Arithmetic utilities for multiplication, division, and square root were accessed by nine separate AN/UYK-20 instructions. Indirection routines, which were called by the RX format routines, emulated the AN/UYK-20 cascaded addressing capabilities. A general purpose move subroutine permitted up to 256 cells of main memory to be moved from one location to another. This routine was used by the emulator's error diagnostic utilities, as well as the load and store multiple address register instructions.

F. INPUT/OUTPUT CONTROLLER

Although the AN/UYK-20's IOC was not emulated, several of its design features were imitated in creating a general purpose input/output controller for the emulator. The input/output command instruction (35 RR) initiated the I/O sequence, and reserved several cells in the memory mapping for I/O control words (Figure V-9). These control words contained fields which indicated what peripheral device was

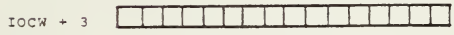
AN/UYK-20 EMULATION

INPUT/OUTPUT COMMAND WORDS



BUFFER ADDRESS
THE MICROCODE ALTERS THE IOCW + 1
DURING BUFFER TRANSFER

BUFFER SIZE
USED FOR CRT OUTPUT



NUMBER OF SECTIONS
USED FOR DISK TRANSFERS (NOT IMPLEMENTED)

* CODE: 00 - CRT
01 - PRT (OUTPUT)/CRD RDR (INPUT)
10 - DISK (NOT IMPLEMENTED)

Figure V-9

selected, the number of buffers that would be passed, whether input or output was desired, and where the buffer was located. Two additional words were reserved for control data required to interface with the disk and the CRT. The disk input/output microcode was not incorporated into the controller, but the framework was provided so the IOC could be readily inserted in the future.

The emulator's IOC section was located in the utility section of the emulation, and operated asynchronously with the Burroughs IOP. Since the IOC was collocated with the emulator in the same D-machine, it was not capable of independent processing. Once it was initiated, emulation execution waited until input/output was completed. The emulation had to use the Burroughs Common Language, since the IOP was microprogrammed to accept only this character set from other D-machines.

In order to transfer a 33-word line printer buffer to the IOP, the AN/UYK-20 emulator IOC had to use 66 AN/UYK-20 words. In order to send 20 words to the CRT, the IOC had to contruct a 40-word buffer. The COMPRES subroutine packed an AN/UYK-20 buffer of 66, 16-bit words into a 32-bit, 33-word buffer, suitable for output to the IOP.

Similarly, on input, a 20-word IOP buffer from the card reader or CRT expanded into a 40-word AN/UYK-20 buffer. In this case, the EXPAND subroutine split every packed 32-bit word of the IOP's buffer into two 16-bit words, suitable for processing by the emulator. These additional data transformations were required because of the buffering requirements of the Burroughs IOP.

VI. EMULATION TESTING

A. METHOD OF TESTING

The fundamental testing technique utilized throughout the development of the emulation consisted of three distinct phases: 1) each module was independently assembled and debugged until successful assembly was achieved, 2) each program segment was then tested for accuracy of the intended operation, and 3) the composite emulation program was tested under actual program conditions allowing for interaction among several modules. A debugging package, which was developed early in the emulation project, was the primary test vehicle for verifying emulation coding. Development and testing of the loader was, however, completed prior to the creation of the debugger.

Loader testing was accomplished by monitoring the control panel lights and synthetically halting program execution via the insertion of 'WAIT' statements, forcing the desired register to be transferred to 'MIR', and then re-examining the panel lights. This process was extremely tedious and time consuming but proved to be an invaluable tool for recognizing peculiar TRANSLANG constructs and microinstruction execution side effects.

Each module was subjected to an extensive desk checking process in order to reduce trivial assembly errors before running it on the machine. After a successful assembly had been achieved, the code was merged with previously existing assembled modules. The emulator was expanded as more modules were added, with utility routines being incorporated prior to the opcode programs.

Initially, the loader was designed, programmed, and thoroughly tested prior to the microprogramming of any AN/UYK-20 instructions or utilities. With the loader implemented, the emulation of the UYK-20 instruction repertoire could procede with a minimum of loader induced problems.

Independent opcodes and various utility routines were added to the previously assembled programs. The final emulation consisted of a total of 205 opcodes, a set of utility programs, and a workable loader.

In the next phase of testing, every module was subjected to a representative number of test cases which demonstrated how closely they compared to the documented AN/UYK-20 operation. Artificial environments were created to subject every opcode to a variety of situations. Whenever the operation of the opcode disagreed with the documented AN/UYK-20 operation, the opcode's function was thoroughly researched. The opcode's side effects were categorized and questions formulated. Often the answers could only be obtained from the Univac field engineer.

To illustrate the complexity of testing, an add instruction was tested with numerous operand combinations: two positive operands, two negative operands, one of each sign yielding a negative result, opposite signs producing a positive result, and opposite signs producing a zero sum. During each addition operation, the overflow, carry, and condition bits had to be monitored in status register 1 to verify their appropriate setting. This level of detail was achieved with all of the implemented opcodes in order to produce an efficient and accurate emulation.

Throughout the entire testing scenario, which composed 20% of the emulation project, the debugger routine (DUMPREG) provided the necessary information for examining opcode execution. A representative sample debugger output is provided in Appendix D.

DUMPREG permitted a snapshot of both AN/UYK-20 general register stacks, PSW, SR1, and SR2, in addition to the D-machine registers (A1, A2, A3, BR1, AMPCR, MIR) and the Burroughs external clock. DUMPREG possessed sufficient flexibility to be incorporated into the microcode at any point. In the final emulator, however, the debugger is user specified, and will dump the AN/UYK-20 emulator environment either at every instruction fetch and program stop, or when called by a loader control card.

B. SAMPLE TEST PROGRAMS

After subjecting the entire emulation to extensive testing, some representative test programs were developed to demonstrate the feasibility of the emulation and its capabilities and performance as compared to an actual AN/UYK-20. There were two programs which were selected because they incorporated numerous emulation features. The two programs were the solution of simultaneous linear equations by Cramer's rule and generation of prime numbers. It must be noted that streamlined program design was not emphasized but rather utilization of a variety of opcodes and features of the emulation.

The program for solving linear equations contained a total of 28 opcodes, requiring 28 opcode execution cycles and 43 instruction fetches. The program demonstrated all four fundamental mathematical operations, numerous store and load functions, a comparison test and a jump instruction. The capability of performing card reader input and output was added when the emulator IOC was completed.

The prime number program demonstrated 30 instructions which required 116 opcode execution cycles, and 122 instruction fetches. This program illustrated numerous comparison tests, looping structures, several jump instructions, a load multiple instruction, addition, and division. The test programs are included in Appendix E.

C. TEST RESULTS

The performance analysis of the test programs consisted primarily of running numerous test cases, examining the results for accuracy and computing the total time required to execute the emulated AN/UYK-20 programs. The timing of the programs was accomplished by using an external clock available on the IOP. The clock time provided a fairly close representation of the emulation execution time, but it cannot be considered a completely accurate measure since the emulation must interrogate the IOP to retrieve the external clock contents. Approximately 50 microseconds are used when sampling the clock.

The time requirements of the AN/UYK-20 program execution were hand-calculated by summing the published instruction execution times as presented in Ref. 21. The emulation performance ratio (EPR) was computed merely to give an approximate indication of the emulation performance. The EPR is the ratio of measured Burroughs emulation time to the calculated AN/UYK-20 execution time. Two EPR figures are recorded: one with paging implemented and one without. The paging EPR figure is significant only in that it indicates how much additional overhead must be incurred when emulating the paging mechanism of the AN/UYK-20. The required additional execution time was about 26%. Naturally, paging overhead is directly proportional to the number of instruction fetches or memory references performed during a program.

The results of program testing are as follows:

	CRAMER'S RULE	PRIME NUMBERS
Number of Opcodes	28	30
Number of Fetches	43	122
Execution Cycles	28	116
Time w/o paging	5000 usec	13000 usec
Time with Paging	8000 usec	17000 usec
AN/UYK-20 Time	78 usec	193 usec
EPR w/o Paging	64 :: 1	67 :: 1
EPR with Paging	103 :: 1	88 :: 1

These figures represent only approximate comparisons of the two machines. These computations provide an estimate of emulation characteristics. An effective EPR without paging was projected to be 65::1.

VII. SUMMARY AND RECOMMENDATIONS

A. EXPERIENCE WITH HARDWARE

The emulation project provided the unique opportunity of learning about two computer systems. The Burroughs D-machine demanded a detailed knowledge of hardware operation, as well as a thorough understanding of the microprogramming language, TRANSLANG. The computer architecture and processor capabilities of the AN/UYK-20 had to be investigated and then integrated into the control store memory of the Burrough's D-machine.

On several occasions, hardware malfunctions with the Burroughs equipment prevented normal system operation. The card reader was inoperable for several weeks, and the disk drive unit had to be repaired several times. During the final three weeks of project development, one D-machine ceased to function properly. This restricted emulation testing to the remaining D-machine.

Although these hardware difficulties impeded normal progress of the project, they did not prevent the emulation from successfully being completed. If a hardware problem prevented emulation testing or debugging, other modules were designed and testing was postponed. This permitted continual emulation development regardless of the hardware

status. In addition, considerable time and effort was invested in trouble-shooting hardware malfunctions, so they could be isolated, diagnosed, and repaired.

B. LESSONS LEARNED

The emulation project brought together many different computer science techniques and disciplines which will be useful in future computer science endeavors. A great deal of experience in both computer architecture and operation was gained in two different types of computers. Microprogramming provided new insight into computer design and revealed many potential applications for programmable control store machines.

Since emulations normally require a large development effort, this project had to incorporate judicious system design and project management principles in order for it to be completed successfully. Careful monitoring of critical stages of the emulation, and coordination of the programming effort to meet scheduled requirements, was necessary throughout this research. The small programming team concept proved to work extremely well in this project.

Finally, several software practices were strictly followed that proved to be invaluable in constructing the emulation. First, modular programming succeeded in partitioning the emulation into discrete modules which could be designed, coded, tested, and implemented individually. The

emulation was constructed in segments, using the previously verified modules as a test bed for the new modules being built. Structured programming and prolific documentation were useful in developing each microprogram routine because they permitted each team member to understand the function of the program and how it could be used.

C. EMULATION PROBLEMS

The most significant problem of the emulation was not having had any experience with an AN/UYK-20 and not having anyone readily available with prior AN/UYK-20 experience. This lack of knowledge created some anxiety when attempting to analyze the ramifications of individual opcodes.

The idiosyncrasies of certain opcodes were not made self-evident by the AN/UYK-20 software manuals. Consequently, numerous code modules were redesigned when more detailed information was provided by a UNIVAC field engineer. This proved to be very time consuming, inefficient, and frustrating.

Another emulation difficulty was the lack of working registers in the host machine as compared with the target machine. While the AN/UYK-20 has either 16 or 32 general registers, depending on whether the second stack option is incorporated, the D-machine has effectively only seven workable registers containing 16 bits or more. Therefore, all AN/UYK-20 registers had to be mapped into S-memory which

created much longer register read/write times. This created significant register manipulation problems which in some cases required main memory references for instructions intended to be strictly register-to-register operations. Consequently, increased execution times resulted in a higher emulation performance ratio (EPR), decreasing the overall emulation performance.

D. RESULTS

Emulating the AN/UYK-20 on a Burroughs D-machine required a considerable amount of preparation and planning before any results were realized. An in depth analysis of each computer's architecture and operating characteristics was conducted to insure that an emulation was feasible in the allotted time period. From the inception of the project, the goal of the emulation was to implement a standard AN/UYK-20 processor. This decision was based on the capability of the D-machine and an estimate of how much time would be involved in developing, debugging, and testing the final product. Although this goal was achieved, it was felt more time could have been devoted to testing and verifying emulation operation.

The AN/UYK-20 emulation was a highly complex microprogramming project involving numerous data structures and transfer protocols. A total of 205 AN/UYK-20 instructions were emulated out of nearly 290 instructions in the repertoire (including all IOC, math pac, and clock interrupt

opcodes). AN/UYK-20 diagnostic programs and user programs will establish the validity of the emulator's construction.

E. RECOMMENDATIONS AND FOLLOW-ON TOPICS

Although 205 opcodes have been implemented, tested, and developed into a working emulation, there are still many challenging avenues to pursue in creating the complete emulation package. First, testing is a continuous process and should be performed in conjunction with code optimization. A comprehensive code optimization effort could improve the EPR without sacrificing code readability.

Second, the floating point and 'math pac' options could be implemented. The addition of floating point arithmetic, trigonometric, and hyperbolic functions would significantly strengthen the scientific capabilities of the emulation. This would be an extremely strenuous undertaking but would permit more tactical data applications of the emulation, increasing its value to the Navy.

One area which could be examined is to perform a comprehensive timing analysis between an AN/UYK-20 and the emulation. This could consist of collecting numerous bench mark programs from Navy AN/UYK-20 installations, where performance data could be accurately obtained. These programs could then be run on the emulator, and analyzed with the bench mark results. The feasibility of replacing or substituting emulation host machines for target machines could be

addressed and supported by the timing analysis study.

An emulation of the AN/UYK-20 input/output controller could be incorporated into the emulation without serious difficulty. Since a second D-machine is available, the IOC channel processor instructions could be emulated and inserted into that D-machine's control memory. The independent processing characteristic of the AN/UYK-20 IOC would be fulfilled using this arrangement.

Finally, when the Burroughs system is linked with the computer science department's PDP 11/50, the AN/UYK-20 emulator could be connected to that system's peripheral resources. This would allow future incorporation of an AN/UYK-20 ULTRA assembler and a CMS-2M compiler into the PDP 11/50 system which could then produce machine language object code files for the AN/UYK-20 emulator to execute.

APPENDIX A. AN/UYK-20 EMULATOR USER'S MANUAL

This description is designed to provide sufficient information to operate the AN/UYK-20 emulation program. It is assumed that the informal Burroughs D-machine manual in the Burroughs laboratory will supply adequate power-on instructions and solve any hardware operating difficulties which may arise.

The procedure for utilizing the AN/UYK-20 emulator can be divided into four phases:

1) selection of the necessary loader control cards (JCL).
2) selection of the AN/UYK-20 program instruction set.
3) implementation of phases one and two into the required card or CRT format.
4) actual hardware implementation on the Burroughs D-machine resulting in program execution.

Phase one can be achieved by selecting the desired loader control cards described in Appendix B. The card format is identical to the CRT format, except that the CRT requires the user to <carriage return> at the end of every line of input data.

Phase two is accomplished by creating the AN/UYK-20 program from a subset of the 205 emulated instructions.

Phase three consists of keypuncning the desired JCL and AN/UYK-20 program in the format illustrated in Appendix C.

The resulting job deck will typically be assembled as follows:

```
?SMLOAD TED/UYK20-OBJECT        % loads the emulator into
                                  micromemory
L  00004                        % load the program into
                                  page 4

C  00206 FAULT INTERRUPT-EXECUTION ENDS "

C  00214 NOT IMPLEMENTED-EXECUTION ENDS "

C  00222 DIVIDE OVERFLOW - FAULT ENTERED"

**************************
other desired JCL
**************************

AN/UYK-20 Program

    03,0,a,00                   % priority interrupt
    (a = 00-17 octal)             (mandatory card)

G                               % commence program
                                  execution

M (or M1)                       % machine status (reg. dump
                                  at termination)

E                               % end job card
```

Finally, phase four consists of the entire program deck being loaded into the card reader. It is assumed that the IOP is at address 0015 hex, the selected interpreter is at address 0549 hex, the line printer is 'READY', and the IRQ switch is 'off'.

The IRQ switch is a three-way toggle switch mounted on the right side of the interpreter. In the up position, IRQ is 'on', horizontally it is 'off', and the downward position is used for external functions.

If either the interpreter or the IOP is not at the proper starting address, clear them by depressing the 'CLEAR' button on each unit. If this procedure does not remedy the situation, consult the user's manual in the laboratory.

Upon reading the ?SMLOAD card, the system will load the AN/UYK-20 emulator object program into the micromemory of the selected interpreter. The program address counter (on the interpreter) will be at the beginning of the emulation program, address 0000 hex. At this point, the user can select CRT input and output by placing the IRQ switch to the 'on' (upward) position. If CRT input is not desired, leave the IRQ switch in the 'off' (horizontal) position. Next, force step the interpreter by momentarily depressing the FST button (uppermost push button on side panel of the interpreter). Do not hold in the FST button. This may cause some undesirable side effects.

After depressing the FST button, the AN/UYK-20 program is loaded into S-memory. If the input is expected from the CRT, the user must enter his program from the console. Otherwise, the emulator will request cards from the card reader. Once the program has been loaded and the 'G' card read, program execution begins.

After successful program termination, the program returns to the loader and asks for more input. At this point, the user may terminate his job, or start another load sequence. It should be remembered that the AN/UYK-20 emulator has been designed for monoprogramming execution. It executes one program at a time, but can execute any number of programs in sequential order if desired. When the job is terminated, the D-machine returns to its starting address (0000 hex) and awaits further processing. When the user returns to the start address, the system is effectively 'master cleared' since S-memory will be cleared prior to executing any further jobs. It is not necessary, however, to use the ?SMLOAD card for additional programs or program re-runs because the emulator object code still resides in micromemory.

If the results of program execution are not as anticipated, use of either a 'T' or 'T1' option (trace card) is recommended to provide the user with a fetch-by-fetch program trace with the output to the printer or CRT respectively.

APPENDIX B. LOADER CONTROL CARD FORMATS

CARD COLUMNS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16 - 20
B	R W B		decimal			addr									
C			decimal			addr			character string followed by a "						
D			decimal			addr									data
E															
G			decimal			addr									
I	1 2					reg#									data
L			dec. page num												
M	 1														
P															
R			dec. number												
S	1 2														
T	 1														

Note: All numeric fields are in decimal.

Control Card Identitifier		Description
B	---	The 'B' card is used to implement the breakpoint feature of the emulation. This features allows the user to specify a decimal address in columns 4-8. Column 2 must contain a R or W to breakpoint on a read or write operation respectively. The default condition is breakpoint on both read and write.
C	---	The 'C' card is used to insert character strings into memory. The character string starts in column 9 and continues until terminated by a quote symbol (") or the string reaches column 80. The decimal address where the character string will be written is given in columns 4-8. A blank or zero address field causes the character string to be inserted at the current load address.
D	---	The 'D' card is used to store decimal data from columns 16-20 into the memory address found in columns 4-8. A blank or zero address field causes the data to be inserted at the currrent load address.
E	---	The 'E' card is used to indicate the end-of-job and therefore is a mandatory control card for job separation.
G	---	The 'G' card or 'GO' card is used to start execution. The starting decimal address is in columns 4-8. The default value of a blank or zero address field will cause program execution to start address 01024.

I --- The 'I' card or set index register card is used to store decimal data from columns 16-20 into the register designated in columns 7-8, into the general register stack (1 or 2) specified in column 2.

L --- The 'L' or load card is used to partition memory into 256, 256-word (32-bit) pages and to load the program into the decimal page as referenced by columns 4-8. Pages 0-3 should not be used because they contain the required emulation register mapping and established workspace. The 'L' is a required control card, and it is recommended that it be the first JCL card.

M --- The 'M' card or machine status card provides a register dump, whereever it is inserted. It is normally placed between the 'G' and 'E' cards in the JCL deck. If a 1 is placed in column 2, the machine status dump will be sent to the CRT. This dump will contain the current value all AN/UYK-20 general registers, the PSW, SR2, next instruction address, the breakpoint address, and clocktime.

P --- The 'P' card is used to implement paging.

R --- The 'R' or reserve space card is used to reserve memory space as specified decimally in columns 4-8.

S --- The 'S' card is used to simulate setting of the program stop switches (1 and 2) on the AN/UYK-20 maintenance panel. Column 2 must contain a 1 or a 2. Two cards are required if both switches are to be set.

T --- The 'T' or trace card provides a fetch-by-fetch program trace, dumping the entire machine status on every fetch cycle. If a 1 is in column 2, the trace will be displayed on the CRT. This card is recommended for debugging programs.

APPENDIX C. AN/UYK-20 EMULATOR INSTRUCTION FORMAT

Format	Card Columns
	5 6 7 8 9 10 11 12 13 14
RR, RL, RI Type 2	opcode , 'f' , ' a ' , ' m '
	5 6 7 8 9 10 11 12
RI Type 1	opcode , 'f' , ' d '
	5 6 7 8 9 10 11 12 13 14 15 16 - 20
RK, RX	opcode , 'f' , ' a ' , ' m ' , ' y '

Notes:

1) All fields are in octal with the exceptions of addresses and the 'y' field which are decimal
2) All fields must be zero-filled
3) The following field restrictions must be followed:

Field	Range	Base
Opcode	00-76	octal
'f'	0-3	octal
'a'	00-17	octal
'm'	00-17	octal
'd'	000-377	octal
'y'	00000-65535	decimal
addr	00000-65535	decimal

APPENDIX D. SAMPLE DEBUGGER OUTPUT

This appendix provides a sample program output illustrating the Trace option. The debugger is called at the beginning of every instruction fetch, and at the termination of the program. Space has been provided for dumping 48 memory addresses including emulated AN/UYK-20 registers and D-machine registers. Each output line consists of eight regions printed in hexadecimal with a total of eight hex digits (32 bits) per region.

The listing is annotated to indicate the identity of each output cell. A summary of cell definitions (numbering from left to right, top to bottom) follows:

DECIMAL ADDRESS	DESCRIPTION
0 - 15	General Register Stack 1
16 - 31	General Register Stack 2
32	Program Status Word (PSW)
33	Breakpoint Register (BREAKPT)
34	Status Register 2 (SR2)
35	NEXTINSTR (next load address)
36	Clocktime
37	Real-Time Clock (RTC)
38	AMPCR
39	A1
40	A2
41	A3
42	MIR
43	BR1
44 - 47	Unused

```
        L   00004
        C   00236 FAULT INTERRUPT-EXECUTION ENDS '
        C   00214 NOT IMPLEMENTED-EXECUTION ENDS '
        C   00222 DIVIDE OVERFLOW - FAULT ENTERED'
        T                           ENABLE DEBUGGER TRACE TO PRT
        P                           ENABLE PAGING
        B   01025                   BREAKPOINT (R/W) AT ADDRESS 1025 DECIMAL
        I1     31       00015
        I1     32       00010
             02,0,02,01             MAKE REG 2 NEGATIVE
             22,0,01,02             ADD REG 2 TO REG 1 AND STORE RESULT IN REG 1
             03,0,C0,00             EXECUTIVE RETURN (HALT)
        G                           COMMENCE EXECUTION
0C0D0D00  0C300D3F  00300C0A  00000000  0C000000  00000000  00C0C000  0C000C00
0C000030  0C300030  00300000  0C0C0C30  0C000000  00000000  0C000C00  0C000C00 (15)
0C000000  0C300C3C  00300C0C  00000000  C0000C00  00000C00  0C000C00  00000C00
0C000000  00300030  00300C00  0C000030  0C0C0000  00000000  0C000C00  00C00C0C (31)
0C700400  0C300401  80300000  0C000403  00000C00  0C000C00  00000C3A  0C7C0400
0C000672  30300030  00320000  0C000032  00000000  0000C030  C0000000  0C000C00 (47)



0C000C00  00300030  00300FF6  00000C00  00000000  00000000  C0000000  0C000C00}GEN REG STACK 1
0C000000  00000030  00300C00  00000030  C0000C00  00000000  0C000C00  0C000C00}
0C000030  00300030  00300000  00000000  0C000C00  00000000  00000C00  00000C00}GEN REG STACK 2
0C000000  0C300030  00300C00  00000000  00000000  00000000  00000C00  00000C00}
0070040C  0C300401  80300C00  0C000403  CC000000  00000C00  0F00C63A  C1F0C401
0C000672  00300821  00300FF6  00000002  00000000  00000000  00000C00  0C000C00



0C000000  0C300035  00300FF6  0C000000  0C000C00  00000000  0C000C00  0C000C00
0C000000  0C300030  00300006  0C000000  00000C00  00000000  0C000C00  00000C00
0C000000  00000030  00300000  00000000  0C000C00  00000C00  00000C00  00000C00
0C000C00  0C300C30  00300000  00000000  0C000000  00000000  0C000C00  0C000C00
CC700400  00300401  80300000  0C000403  00000000  00300C00  0C00063A  10F0C402
0C000672  0C314812  00300005  00000002  CC000C00  0C000000  00000000  00000000
PSW       BREAKPT   SR2       NEXTINSTR CLOCKTIME  RTC       AMPCR     A1

0C000404  00300035  00300FF6  0C000030  0C000000  00000000  00000000  0C000C00
0C000000  0C300030  00300C00  00000000  C0000C00  00000000  0C000C00  0C000C00
0C000C00  0C300C30  00300000  00000000  0C000000  0000C00C  00000000  00000C0C
0C000C00  00300000  00300000  0C000000  CC000030  00000000  00000000  00000C00
0C700400  00300401  80300000  0C000403  0C003E2C  00000000  0C0C05E4  10F0C403
0C000000  0C300030  00303E2C  00000032  0F0C0000  00000C00  00000000  0C000C0C
A2        A3        MIR       BRI       |<------------ 44 - 47 ------------>|

        E                           END OF JOB
```

APPENDIX E. SAMPLE TEST PROGRAMS

The programs listed in this appendix were designed to illustrate how the AN/UYK-20 emulator can be used for developing programs. The first two programs presented are the generation of prime numbers and the solution of simultaneous linear equations by Cramer's Rule. They depict several AN/UYK-20 control structures such as looping, iteration, and condition code testing as well as numerous load, store, and arithmetic operations. The last program performs the Cramer's Rule algorithm and demonstrates input/output routines.

```
L  00034                        PRIME NUMBER GENERATOR
11                              PRIME NUMBERS RETURNED IN GEN. REGISTERS
P                               THIS CARD WILL ENVOKE PAGING
C  00206 FAULT INTERRUPT-EXECUTION ENDS '
C  00214 NOT IMPLEMENTED-EXECUTION ENDS '
C  00222 DIVIDE OVERFLOW - FAULT ENTERED'
    63,0,04,01                  LITERAL LOAD, 1 INTO REG 4 (1ST PRIME)
    63,0,05,02                  LITERAL LOAD, 2 INTO REG 5 (2ND PRIME)
    63,0,06,02                  LITERAL LOAD, 2 INTO REG 6 (COUNTER)
    63,0,07,05                  LITERAL LOAD, NO. OF PRIMES INTO REG 7
    11,3,04,00,02048            STORE  A 1 INTO ADDR 2048
    11,3,05,00,02049            STORE  A 2 INTO ADDR 2049
    24,0,07,04                  COMPARE N (NO. OF PRIMES) & 1
    44,1,026                    LOCAL JUMP, IF EQUAL
    24,0,07,05                  COMPARE N AND 2
    44,1,023                    LOCAL JUMP, IF EQUAL
    63,0,11,02                  LITERAL LOAD, 2 INTO REG 11 (CANDIDATE)
    62,2,11,01                  LITERAL ADD, INCREMENT CANDIDATE
    63,0,17,02                  LITERAL LOAD, 2 INTO REG 10 (TRIAL DIVIS)
    01,0,13,11                  LOAD REG, CANDIDATE INTO REG 13
    01,0,12,00                  LOAD REG 12 WITH 0
    27,0,12,10                  DIVIDE REG, CANDIDATE BY TRIAL DIVISOR
    24,0,09,12                  COMPARE, REMAINDER WITH 0
    44,1,371                    JUMP IF EQUAL, NON-PRIME, GET NEXT CAND.
    62,2,13,01                  LITERAL ADD, INCREMENT TRIAL DIVISOR
    24,0,10,11                  COMPARE DIVISOR WITH CANDIDATE
    47,1,370                    JUMP IF LESS THAN, TRY NEW DIVISOR
    11,3,11,06,02048            STORE FOUND PRIME INTO MEMORY
    62,2,06,01                  LITERAL ADD, INCREMENT PRIME COUNTER
    24,0,05,07                  COMPARE, PRIME COUNTER WITH N
    47,1,351                    JUMP IF LESS THAN, GET NEXT PRIME CAND.
    03,3,03,04,02048            LOAD MULT, PRIMES FROM MEMORY INTO REGS
    03,0,17,00                  PRIORITY INTERRUPT, HALT EXECUTION
    63,0,01,02                  LIT LOAD, FORCE FEED 2 INTO REG 01
    63,0,07,01                  LIT LOAD, FORCE FEED 1 INTO REG 00
    03,0,17,00                  PRIORITY INTERRUPT, HALT EXECUTION
G                               GO, COMMENCE EXCUTION
M                               MACHINE STATUS TO THE PRINTER
00000001  00300032  00300003  00000005  00000007  00000002  00000005  00000005
00000007  00300037  00300001  00000001  00000000  00000000  00000000  00000420
00000000  00300030  00300000  00000000  00000000  00000000  00300000  00000000
00000030  00300030  00300000  00300000  00000000  00000000  00000000  00000000
00400400  00300030  00300000  00000422  00300018  00000030  00000161  00000000
00000030  00300030  00300000  00000005  00000000  00000000  00000000  00000000

E                               END OF JOB
```

```
L  00034                        LINEAR EQUATIONS BY CRAMER'S RULE.
T1                              X IS RETURNED IN REG 01, Y IN REG 05
C  00206 FAULT INTERRUPT-EXECUTION ENDS '
C  00214 NOT IMPLEMENTED-EXECUTION ENDS '
C  00222 DIVIDE OVERFLOW - FAULT ENTERED'
D  02048        00030           BOTH EQUATIONS MUST BE IN SID FORM
D  02049        00030           A (Y) PLUS B (X) EQUALS C
D  02050        00003           Y COEFF OF 1ST EQN
D  02051        65531           X COEFF OF 1ST EQN
D  02052        00019           CONSTANT OF 1ST EQN
D  02053        00002           Y COEFF OF 2ND EQN
D  02054        65532           X COEFF OF 2ND EQN
D  02055        00016           CONSTANT OF 2ND EQN
    01,3,01,00,02050            STORE X COEFF, EQN 1, INTO REG 01
    01,3,03,00,02051            STORE Y COEFF, EQN 1, INTO REG 03
    01,3,05,00,02052            STORE CONSTANT, EQN 1, INTO REG 05
    01,3,11,00,02053            STORE Y COEFF, EQN 2, INTO REG 11
    01,3,13,00,02054            STORE X COEFF, EQN 2, INTO REG 13
    01,3,15,00,02055            STORE CONSTANT, EQN 2, INTO REG 15
    26,0,00,13                  MULT LEFT DIAG OF DENOM, RESULT IN 00/01
    26,0,10,03                  MULT RT.  DIAG OF DENOM, RESULT IN 10/11
    21,0,00,10                  SUBTRACT DOUBLE, RESULT OF DENOM IN 00/01
    25,3,00,17,02048            COMPARE DENOM WITH ZERO
    94,1,052                    JUMP, IF EQUAL (ZERO DETERMINANT)
    12,3,03,06,00014            STORE DOUBLE, DENOM INTO 16/17
    01,2,C0,00,60030            LOAD 00 INTO REG C0
    01,3,01,00,02050            RESTORE Y COEFF, EQN 1, INTO REG 01
    01,3,03,00,02051            RESTORE X COEFF, EQN 1, INTO REG 03
    01,3,05,00,02052            RESTORE CONSTANT OF EQN 1 IN REG 05
    01,3,11,00,02053            RESTORE Y COEFF OF EQN 2 IN REG 11
    01,3,13,00,02054            RESTORE X COEFF OF EQN 2 IN REG 13
    01,3,15,00,02055            RESTORE CONSTANT OF EQN 2 IN REG 15
    26,0,C9,15                  X NUM LEFT DIAG MULT, RESULT IN 00/01
    26,0,15,05                  X NUM RT.  DIAG MULT, RESULT IN 10/11
    21,0,00,10                  X NUM RESULT IN REG 00/01
    27,0,00,17                  X RESIDES IN REG C1
    26,0,C9,13                  Y NUM LEFT DIAG MULT, RESULT REG 04/05
    26,0,19,03                  Y NUM RT.  DIAG MULT, RESULT REG 14/15
    21,0,C9,19                  DOUBLE SUBTRACT (FVAL NUM), RESULT 04/05
    27,0,09,17                  DIVIDE NUM/DENOM, Y RESIDES IN 05
    03,0,17,00                  PRIORITY INTERRUPT, HALT EXECUTION
G                               GO, COMMENCE EXECUTION
M                               MACHINE STATUS TO THE PRINTER
0C000000  0C30FFF8  00300000  0C00FFF8  C0000000  0000FFFE  00000C00  0C000C00
0C000800  00000026  00300000  0C00FFFC  0C00FFFF  0000FF80  0C00FFFF  00000420
0C000000  00300000  00300000  0C000030  0C000000  00000C00  00000C00  00000C00
0C000000  00300000  00300000  00000030  00000030  00000000  00000C00  00000C00
0C000400  00300320  00300000  00000428  00000005  00000800  00000161  00000C00
0C000030  00300C50  00300C06  0C300004  00000000  00000000  0C0C0C30  00000C00

F                               END OF JOB
```

```
P                              ****LINEAR EQUATIONS BY CRAMER'S RULE.****
L   00034
C   00236 FAULT INTERRUPT-EXECUTION ENDS '
C   00214 NOT IMPLEMENTED-EXECUTION ENDS '
C   00222 DIVIDE OVERFLOW - FAULT ENTERED'
I2      30          17654               IOCW FOR 20 BUFFERS FROM CARD READER
I2      31          04076
I2      32          16758               IOCW FOR 6 BUFFERS FROM CARD READER
I2      33          05120
I2      34          16735               OUTPUT IOCW FOR 5 BUFFERS TO PRT
I2      35          08076
I2      36          17655               OUTPUT 20 BUFFERS TO PRT
I2      37          10240
I2      38          12356               REGISTER CONTAINING TWO BCL SPACE CHARACTERS
I2      11          000040
I2      15          000030
D   02048           000030              BOTH EQUATIONS MUST BE IN STD FORM
D   02049           000030              A (Y) PLUS B (X) EQUALS C
D   02050           000001              Y COEFF OF 1ST EQN
D   02051           000001              X COEFF OF 1ST EQN
D   02052           000033              CONSTANT OF 1ST EQN
D   02053           000001              Y COEFF OF 2ND EQN
D   02054           65535               X COEFF OF 2ND EQN
D   02055           65535               CONSTANT OF 2ND EQN
        63,0,13,01                      LOAD 1 INTO REG. 13
        03,0,13,05                      LOAD SR1 WITH REG 13
        12,1,00,17                      STORE DOUBLE INTO IOCW
        35,0,03,00                      INPUT 14 BUFFERS FROM CARD READER
        12,1,02,17                      STORE DOUBLE INTO IOCW
        35,0,03,00                      INPUT 6 BUFFERS FROM THE CARD READER
        01,2,12,00,001320               INITIALIZE BUFFER WORDS COUNT (20 PRT BUFFERS)
        01,2,11,00,10240                INITIALIZE OUTPUT BUFFER ADDR
        15,1,10,11                      STORE (RA) INTO Y* AND INDEX BY 1 (STORE SPACES)
        02,0,12,11                      DECREMENT BUFFER COUNTER
        45,1,375                        COUNTER EQUAL ZERO
        01,2,10,00,040076               INITIALIZE BUFFER ADDRESS OF INPUT
        01,2,15,00,000020               BUFFER COUNT
        01,2,11,00,10240                INTIALIZE DESTINATION BUFFER ADDR
        05,1,15,10                      LOAD AND INDEX BY 1
        15,1,15,11                      STORE RA INTO Y* AND INDEX Y*
        02,0,13,11                      DECREMENT COUNTER
        45,1,374                        COUNTER EQUAL ZERO
        22,2,11,00,000026               ADD 26 TO BUFFER ADDR
        01,2,13,00,000040               RESET  BUFFER WORD COUNT
        02,0,13,11                      DECREMENT NEXT COUNTER (NO. OF BUFFERS)
        45,1,356                        COUNTER EQUAL ZERO
        12,1,05,17                      STORE DOUBLE INTO IOCW
        35,0,03,00                      OUTPUT 14 BUFFERS TO PRT
```

```
63,0,15,01              ***RE-ENTRY POINT FOR REPETITIVE EXECUTION***
03,0,15,05                 LOAD SR1 WITH A 1 FROM REG 13
01,2,11,00,08096           INITIALIZE NEXT OUTPUT BUFFER ADDRESS
01,2,12,00,00330           INITIALIZE NEXT BUFFER WORD COUNT (5 PRT BUFFERS)
01,2,13,00,00035           BUFFER COUNT
01,2,14,00,05120           INITIALIZE  BUFFER ADDR OF SECOND INPUT BUFFER
15,1,10,11                 STORE (RA) INTO Y* AND INDEX BY 1 (STORE SPACES)
02,0,12,11                 DECREMENT BUFFER COUNTER
45,1,375                   COUNTER EQUAL ZERO
01,2,11,00,08096           REINITIALIZE BUFFER ADDR IN REG 9
05,1,15,14                 LOAD AND INDEX BY 1
15,1,15,11                 STORE RA INTO Y* AND INDEX Y*
02,0,13,11                 DECREMENT COUNTER
45,1,374                   COUNTER EQUAL ZERO
22,2,11,00,00026           ADD 26 TO BUFFER ADDR
01,2,13,00,00040           RESET  BUFFER WORD COUNT
02,0,13,11                 DECREMENT NEXT COUNTER (NO. OF BUFFERS)
45,1,366                   COUNTER EQUAL ZERO
63,0,15,00                 LOAD REG 13 WITH 0
03,0,15,05                 LOAD SR1 WITH 0
01,3,01,00,02030        ***CRAMER'S RULE ALGORITHM***  STORE X OF EQN 1
01,3,03,00,02031           STORE Y COEFF, EQN 1, INTO REG 03
01,3,05,00,02032           STORE CONSTANT, EQN 1, INTO REG 05
01,3,11,00,02033           STORE Y COEFF, EQN 2, INTO REG 11
01,3,13,00,02034           STORE X COEFF, EQN 2, INTO REG 13
01,3,15,00,02035           STORE CONSTANT, EQN 2, INTO REG 15
26,0,00,13                 MULT LEFT DIAG OF DENOM, RESULT IN 00/01
26,0,17,03                 MULT RT.  DIAG OF DENOM, RESULT IN 10/11
21,0,00,1C                 SUBTRACT DOUBLE, RESULT OF DENOM IN 00/01
25,3,00,17,02048           COMPARE DENOM WITH ZERO
44,1,052                   JUMP, IF EQUAL (ZERO DETERMINANT)
12,3,00,06,00014           STORE DOUBLE, DENOM INTO 16/17
01,2,00,00,00030           LOAD 00 INTO REG 00
01,3,01,00,02050           RESTORE Y COEFF, EQN 1, INTO REG 01
01,3,03,00,02051           RESTORE X COEFF, EQN 1, INTO REG 03
01,3,05,00,02052           RESTORE CONSTANT OF EQN 1 IN REG 05
01,3,11,00,02053           RESTORE Y COEFF OF EQN 2 IN REG 11
01,3,13,00,02054           RESTORE X COEFF OF EQN 2 IN REG 13
01,3,15,00,02055           RESTORE CONSTANT OF EQN 2 IN REG 15
26,0,00,15                 X NUM LEFT DIAG MULT, RESULT IN 00/01
26,0,19,05                 X NUM RT.  DIAG MULT, RESULT IN 10/11
21,0,00,1C                 X NUM RESULT IN REG 00/01
27,0,00,17                 X RESIDES IN REG 01
26,0,04,13                 Y NUM LEFT DIAG MULT, RESULT REG 04/05
26,0,14,03                 Y NUM RT.  DIAG MULT, RESULT REG 14/15
21,0,04,14                 DOUBLE SUBTRACT (EVAL NUM), RESULT 04/05
27,0,04,17                 DIVIDE NUM/DENOM, Y RESIDES IN 05
```

```
01,2,07,00,16339        ***INT TO BCL CONVERSION*** TWICE BUFFER ADDR PLUS 1
63,0,03,00                 LIT LOAD, C INTO REG 0
63,3,00,12                 LIT DIVIDE, REG 0/1 BY 10
10,3,03,07,00030           BYTE STORE, REG 7 MUST CONTAIN TWICE ADDR
44,1,033                   LOCAL JUMP EQUAL (ZERO QUOTIENT)
63,0,07,00                 LIT LOAD, C INTO REG 0
62,0,07,01                 LIT SUBTRACT, 1 FROM REG 7
40,1,371                   LOCAL JUMP
01,2,07,00,16633           LOAD CONSTANT, TWICE ADDR PLUS 1 IN REG. 7
63,0,04,00                 LIT LOAD, C INTO REG 4
63,3,04,12                 LIT DIVIDE, REG 4/5 BY 10
10,3,04,07,00030           BYTE JUMP
44,1,033                   JUMP EQUAL, CHECK FOR ZERO QUOTIENT
63,0,04,00                 LIT LOAD, C INTO REG 4
62,0,07,01                 LIT SUBTRACT, 1 FROM REG 7
40,1,371                   LOCAL JUMP
63,0,15,01                 LOAD REG 13 WITH A 1
03,0,15,05                 LOAD SR1 WITH A 1
12,1,04,17                 STORE DOUBLE INTO IOCW
35,0,03,00                 DUMP 5 BUFFERS TO PRT
03,0,06,00                 HALT
63,0,15,01              ***NO SOLUTION CASE***
03,0,15,05                 LOAD SR1 WITH A 1
20,2,11,00,00178           RECOMPUTE OUTPUT BUFFER ADDR FOR 'NO SOLUTION'
01,2,13,00,00040           REINITIALIZE COUNTER
05,1,15,14                 LOAD AND INDEX BY 1
15,1,15,11                 STORE RA INTO Y* AND INDEX Y*
02,0,13,11                 DECREMENT COUNTER
45,1,374                   COUNTER EQUAL ZERO
12,1,04,17                 STORE DOUBLE INTO IOCW
35,0,03,00                 DUMP 5 BUFFERS TO PRT
03,0,06,00                 HALT
```

```
G                               COMMENCE EXECUTION


THIS IS A DEMONSTRATION PROGRAM OF THE AN/JYK20 EMULATOR.
THE PROGRAM SOLVES TWO SIMULTANEOUS EQUATIONS BY CRAMER'S RULE.


THE INPUT COEFFICIENT VALUES AND CONSTANTS FOR EACH EQUATION MUST BE INSERTED
USING LOADER CONTROL CARDS IN THE FOLLOWING LOCATIONS

1ST EQUATION.   A1 IN LOCATION 2050 D.
                B1 IN LOCATION 2051 D.
                CONSTANT IN LOCATION 2052 D.

2ND EQUATION.   A2 IN LOCATION 2053 D.
                B2 IN LOCATION 2054 D.
                CONSTANT IN LOCATION 2055 D.


     THE RESULT IS.
     X >        2

     Y >        1
TO CHANGE VARIABLES INSERT LOADER JCL HERE, OTHERWISE EXIT
        I2    L1        00030           REINITIALIZE BUFFER WORD COUNT FOR BUFFER INIT.
        D  02053        00031           Y COEFF OF 2ND EQUATION
        D  02054        00031           X COEFF OF 2ND EQUATION
        D  02055        00033           CONSTANT OF 2ND EQUATION
        G  00031                        RE-EXECUTE PROGRAM AT ADDR 1055 DECIMAL
     THE RESULT IS.
     X >
        NO SOLUTION
     Y >
TO CHANGE VARIABLES INSERT LOADER JCL HERE, OTHERWISE EXIT
        E                               END EXECUTION
```

APPENDIX F. EMULATOR LISTING

This appendix provides a listing of the TRANSLANG assembler output of the AN/UYK-20 emulation. A source file copy of the emulator exists on disk as well as on cards. A microinstruction object file is also maintained on disk.

The listing is divided into four sections. The left most section contains the microinstruction address composed of four hexadecimal digits followed by a 56-bit microinstruction created from a TRANSLANG instruction. The center section contains the TRANSLANG source which includes labels, an instruction, and/or a comment field. The final number printed on the right most side of the listing is the sequence number of the TRANSLANG source statement. This number, printed in decimal, is created by a Burroughs software utility program called CARD-LIST. This number must be used when editing source programs on disk.

```
                        %MERGE TED/UYK20
                        PROGRAM TED/UYK20-OBJECT                                                    0C0C1C00 D
                        % * * * * * * * * * * * * * MAIN MEMORY ADDRESS MAPPING * * * * * * * * * * 00002C00 D
                                                    %                                               0000 3C00 D
                        GREG1 = 0                   % ADDRESS OF GENERAL REGISTER STACK ONE          00004C00 D
                        GREG2 = 16                  % ADDRESS OF GENERAL REGISTER STACK TWO          0C005000 D
                        PSW = 32                    % PROGRAM STATUS WORD (PAR & SR #1)              0000 6C00 D
                        BKPT = 33                   % BREAKPOINT REGISTER                            00007C00 D
                        STATUS2 = 34                % STATUS REGISTER TWO                            CCCC8C00 D
                        NXTINSTR = 35               % ADDRESS OF NEXT INSTRUCTION                    0C0C9000 D
                        CLOCKTIME = 36              % TIME OF EXECUTION                              00010C00 D
                        RTC = 37                    % REAL TIME CLOCK                                00011C00 D
                        WORKSPACE = 38              % TEMPORARY STORAGE STACK                        0C012C00 D
                        STACK = 44                  % TEMPORARY STORAGE STACK                        0C013C0C D
                        IOCW = 50                   % INPUT OUTPUT COMMAND WORDS (THRU 53)           0C014C0C D
                        HEXADDR = 120               % HEX ADDR TO BE PRINTED (TWO WORD VAL)          00015C00 D
                                                    %                                               00016C00 D
                                                    % CARDBUFFER ADDRESSES                           00017C00 D
                                                    %                                               0C018C00 D
                        CARDBUF = 122               % ADDRESS OF CARD BUFFER                         0C019C00 D
                        CARD1X4 = 122               % ADDRESS OF CARD COLS 1-4                       0C020C00 D
                        CARD5X8 = 123               % ADDRESS OF CARD COLS 5-8                       00021C00 D
                        CARD9X12 = 124              % ADDRESS OF CARD COLS 9-12                      0C022C00 D
                        CARD13X16 = 125             % ADDRESS OF CARD COLS 13-16                     00023C00 D
                        CARD77X80 = 141             % ADDRESS OF CARD COLS 77-80. END OF CARD        00024C00 D
                                                    %                                               00025C00 D
                        PRINTBUFF = 153             % PRINT BUFFER LOCATION                          0C026C00 D
                                                    %                                               00027C0C D
                        CRTBUFF = 186               % CRT BUFFER (20 WORDS)                          00028C00 E
                                                    %                                               0C029C00 D
                        ERRORLIST = 206             % LIST OF POSSIBLE ERRORS                        0002CC00 D
                        PAGEREG = 768               % PAGE ADDRESS REGISTERS STACK BASE              00031C00 D
                                                    % (0-255)                                       00032C00 D
                                                    %                                               00033C00 D
                        % * * * * * * * * * * * * * * LOADER * * * * * * * * * * * * * * * * * * * 00034000 D
                                                    %                                               0C035C00 D
                                                    % BEGIN PROGRAM BY ZEROING MAIN MEMORY          00036C00 D
                        START:                      % RESTART RETURN POINT                           00037C0C D
                                                    % IF CRT INPUT DESIRED, PUT IRO SW ON           00038C00 D
0C00  4800 C00D 0030 C0F0           WAIT            % TOGGLE EXECUTION BY SINGLE STEP                00039C0C D
0C01  0D88 00C1 401C 00F0           SET GC1; WHEN GC1 THEN 1 L = A1,MAR2                            0004 0C00 D
0C02  0000 000D 0030 C020           COMP 16 = SAR                                                   0C041000 D
0C03  4812 00C3 003C 00F0           0 = MIR; SAVE                                                   0C042C00 D
0C04  9809 ACDE 409C 1CF0           MW2; A1 -1 = A1; IF SA1                                         0C043C00 D
0C05  6819 000D 0030 C0FC           IF ABT THEN SKIP ELSE STEP  % WHEN A1 = ZERO 64K OF MEMORY = 0  0C044C00 D
0C06  9C28 A0C3 001C 00FC           WHEN SA1 THEN A1 = MAR2; JUMP                                   0C045C00 D
0C07  0000 000D 0030 C040           NEWCARD - 1 = MPCR  % FETCH A CARD                              0C046C0C D
                                                        %                                           0C047000 D
                                                        %                                           0C048C00 D
                                                        %                                           0C049C00 D
                        CONTROLCD:                      % THIS ROUTINE WILL READ COLUMNS 1-4 AND     0C050C0C D
                                                        % EXAMINE COLUMN 1 FOR A CONTROL CARD       0C051C0C D
                                                        % IDENTIFIER.  IF THE CARD IS A CONTROL      0C052C00 D
                                                        % THE APPROPRIATE ROUTINE WILL BE CALLED.    0C053C00 D
0C08  4809 20C3 001C C0F0           LIT = MAR2          % LOAD ADDRESS OF CARD COLUMNS 1-4.          0C054C00 D
0C09  07AC 00C0 0030 C0EC           CARD1X4 = LIT                                                   0C055C00 D
0C0A  D010 00C0 0C30 0060           INPUT - 1 = CPCR    % READ COLUMNS 1-4                          0C056C00 D
0C0B  4809 20C1 C030 40F0           LIT C = A1,CSAR     % LOAD MSB OF A1 WITH 00111111              0C057C00 D
```

```
0F0C  03FC 0000 0000 0090          63 = LIT; B = SAR                                    00058C00 D
0C0D  4809 AC56 8B3C 00F0          A1 AND B R = B              % FILL LSB OF B REGISTER WITH CHARACTER  00059C00 D
                                                               % FROM COLUMN ONE.       00060C00 D
                                                               %                        00061C00 D
0C0E  4809 2C52 0030 00F0          LIT EQL B                                            00062C00 D
0C0F  012C 0000 0030 00E0          18 = LIT                    % DCL CODE FOR "B"        00063C00 D
0C10  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "B"  00064000 D
0C11  002C 0003 0030 0040          BREAKPOINT - 1 = MPCR       % JUMP TO BREAKPOINT ROUTINE  00065C00 D
                                                               %                        00066C00 D
0C12  019C 0003 0030 00E0          29 = LIT                    % DCL CODE FOR "D"        00067C00 D
0C13  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "D"  00068C00 D
0C14  003C 0003 0030 0040          DATA - 1 = MPCR             % JUMP TO DATA ROUTINE    00069C00 D
                                                               %                        00070CC0 D
0C15  0150 0003 0030 00E0          21 = LIT                    % DCL CODE FOR "E"        00071C00 D
0C16  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "E"  00072C00 D
0C17  FFFC 0003 0030 0050          START - 1 = MPCR            % JOB TERMINATION. REINITIALIZE MEMORY  00073000 D
                                                               %                        00074C00 D
0C18  0170 0003 0030 00E0          23 = LIT                    % DCL CODE FOR "G"        00075C00 D
0C19  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "G"  00076C00 D
0C1A  004C 0003 0030 0040          GO - 1 = MPCR               % JUMP TO GO ROUTINE      00077000 D
                                                               %                        00078C00 D
0C1B  0190 0003 0030 00E0          25 = LIT                    % DCL CODE FOR "I"        00079C00 D
0C1C  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "I"  00080C00 D
0C1D  0050 0000 0000 0040          SETINDEXREG - 1 = MPCR  % JUMP TO THE SET INDEX REG ROUTINE  00081000 D
                                                               %                        00082C00 D
0C1E  023C 0003 0000 00E0          35 = LIT                    % DCL CODE FOR "L"        00083C00 D
0C1F  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP   % CHECK FOR "L"  00084C00 D
0C20  006C 0003 0030 0040          LOAD - 1 = MPCR             % JUMP TO LOAD ROUTINE    00085C00 D
                                                               %                        00086C00 D
                                                               %                        00087C00 D
0C21  0270 0003 0030 00E0          39 = LIT                    % DCL CODE FOR "P"        00088C00 D
0C22  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "P"  00089C0C D
0C23  0070 0003 0030 0040          SETPAGING - 1 = MPCR        % JUMP TO SET PAGING BIT ROUTINE  00090C00 D
                                                               %                        00091C00 D
0C24  0290 0003 0030 00E0          41 = LIT                    % DCL CODE FOR "R"        00092C00 D
0C25  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "R"  00093000 D
0C26  0080 0000 0000 0040          RESERVESPACE - 1 = MPCR  % JUMP TO RESERVESPACE ROUTINE  00094000 D
                                                               %                        00095C0C D
0C27  032C 0003 0030 00E0          50 = LIT                    % DCL CODE FOR "S"        00096C00 D
0C28  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "S"  00097C00 D
0C29  0090 0000 0000 0040          SWITCHES - 1 = MPCR         % JUMP TO SWITCHES ROUTINE  00098C00 D
                                                               %                        00099C0C D
0C2A  033C 0003 0000 00E0          51 = LIT                    % DCL CODE FOR "T"        00100C00 D
0C2B  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "T"  00101C00 D
0C2C  00A0 0003 0030 0040          TRACE - 1 = MPCR            % JUMP TO TRACE ROUTINE   00102C00 D
                                                               %                        00103C0C D
0C2D  0130 0003 0030 00E0          19 = LIT                    % DCL CODE FOR "C"        00104C0C D
0C2E  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "C"  00105C0C D
0C2F  00B0 0003 0030 0040          CHARSTRING - 1 = MPCR  % JUMP TO CHARACTER STRING ROUTINE  00106C00 D
                                                               %                        00107C00 D
0C30  030C 0000 0030 00EC          49 = LIT                    % ECL CODE FOR BLANK      00108C0C D
0C31  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR BLANK  00109C00 D
0C32  00C0 0000 0030 0040          INSTRUCTION - 1 = MPCR  % JUMP TO THE INSTRUCTION HANDLING SUBR  00110C00 D
                                                               %                        00111C00 D
0C33  0240 0003 0030 00E0          36 = LIT                    % DCL CODE FOR "M"        00112C00 D
0C34  680B 2C52 0030 00F0          LIT EQL BJ IF TRUE THEN STEP ELSE SKIP  % CHECK FOR "M"  00113C00 D
0C35  0000 0003 0030 0040          MACHSTAT - 1 = MPCR         % JUMP TO MACHINE STATUS ROUTINE  00114C0C D
                                                               %                        00115C00 D
0C36  00EC 0000 0030 0040          NEWCARD - 1 = MPCR          % DEFAULT. GET NEW CARD   00116C00 D
                                                               %                        00117000 D
```

```
                                LOAD:                   % SET UP MEMORY INTO 256 PAGES OF 256     00118000 D
                                                        % WORDS (32 BIT) EACH.  THE PAGE NO. IN   00119000 D
                                                        % THE "L" CARD (COL. 5-8) WILL BE         00120000 D
                                                        % TRANSLATED INTO  THE BASE ADDRESS OF    00121000 D
                                                        % THE SELECTED PAGE AND STORED IN         00122000 D
                                                        % THE NEXTINST REG TO BE UTILIZED AS THE  00123000 D
                                                        % BASE ADDRESS TO LOAD THE PROGRAM.       00124000 D
0037  4809 0000 CB50 00FC            0 = B,MIR           % ZERO B AND MIR                          00125000 D
0038  4809 0643 001C 00F0            AMPCR = MAR2        %SET UP ADDRESS OF PAGE REGISTER         00126000 D
0039  3000 0000 0000 00C0            PAGEREG = AMPCR                                              00127000 D
003A  0FE0 0003 0030 00E0            254 = LIT           % SET UP CTR FOR 255 + 1 ITERATIONS       00128000 D
003B  4812 0003 0C01 00F0            LOTR; SAVE          % THIS LOOP WILL INITIALIZE 256 PAGE ADDR 00129000 D
                                                        % REGISTERS TO THEIR BASE ADDRESSES.      00130000 D
003C  9809 0000 0030 1CF0            MW2; BMI; IF SAI    % TRANSFER MIR TO B AFTER MEMORY WRITE    00131000 D
003D  9008 0046 0032 00F0            WHEN SAI THEN B + 1 = MIR,INC % INCREMENT PAGE NO. TO BE LOADED00132000 D
                                                        % INTO THE NEXT PAGE ADDRESS              00133000 D
                                                        % AND INCREMENT LOOP COUNTER              00134000 D
003E  8429 0F46 001C 00FC            IF NOT COV THEN BMAR + 1 = MAR2; JUMP  % INCREMENT ADDR OF PAGE00135000 D
003F  4809 2003 001C 00F0            LIT = MAR2          % LOAD ADDR OF COL 1-4                    00136000 D
0040  07A0 0003 0030 00EC            CARD1X4 = LIT                                                00137000 D
0041  00F0 0000 003C C060            INPUT - 1 = CPCR    % READ COLS 1-4                           00138000 D
0042  0100 0003 0030 C060            GETADDR - 1 = CPCR  % GET OCTAL VALUE OF PAGE NO. IN COL.6-8 00139000 D
0043  4809 0041 0050 00F0            B L = MIR           % MULT PAGE NO. BY 256 TO GET BASE ADDR   00140000 D
0044  0230 0000 0000 0060            NXTINSTR = LIT; COMP B = SAR  % INTO NXTINSTR REG WHICH WILL  00141000 D
0045  4809 2003 CC1C 00F0            LIT = MAR2          % WRITE THE BASE ADDRESS OF SELECTED PAGE 00142000 D
0046  0110 0003 0030 0040            OUT - 1 = MPCR      % TO BE USED AS THE BASE ADDRESS TO LOAD  00143000 D
                                                        % THE PROGRAM.                            00144000 D
                                                        %                                         00145000 D
                                                        %                                         00146000 D
                                INPUT:                  % THIS ROUTINE WILL READ THE CONTENT OF   00147000 D
                                                        % THE MEMORY ADDRESS IN MAR2.              00148000 D
0047  A809 0003 003C 0CF0            MR2; IF RDC         % PERFORM READ                            00149000 D
0048  AC28 0000 0000 00F0            WHEN RDC THEN BEX; JUMP  % READ CONTENTS OF MAR2 INTO B.      00150000 D
                                                        %                                         00151000 D
                                                        %                                         00152000 D
                                GO:                     % THIS ROUTINE RETRIEVES THE PAR CONTENTS 00153000 D
                                                        % AND STATUS REG 1 CONTENTS AND PACKS     00154000 D
                                                        % THEM INTO A1 AND THEN JUMPS TO THE      00155000 D
                                                        % EMULATOR                                00156000 D
0049  0120 0003 0030 0060            GETADDR - 1 = CPCR  % GET ADDRESS IN COL. 5-8 OF "GO" CARD   00157000 D
004A  4809 0001 4C0C 00F0            1 L = A1                                                     00158000 D
004B  A200 0003 0030 C0A0            COMP 10 = SAR; PSW = LIT                                     00159000 D
004C  4809 A040 4070 00F0            A1 + B = A1         % OFFSET ADDRESS BY 1024                  00160000 D
004D  4809 2003 001C 00F0            LIT = MAR2          % LOAD ADDRESS OF PSW                     00161000 D
004E  0460 0003 0000 C060            INPUT - 1 = CPCR    % RETRIEVE CONTENTS OF PSW                00162000 D
004F  4809 A050 4030 00F0            A1 OR B = MIR, A1   % STORE COMPLETE PSW INTO MIR AND A1     00163000 D
0050  0130 0003 0030 0060            OUTPUT1 - 1 = CPCR  % LOAD PAR REG WITH START ADDRESS OF PROG00164000 D
0051  0140 0003 0030 0040            STARTIME - 1 = MPCR % START EMULATION OF UYK20               00165000 D
                                                        % START CLOCK AND FETCH FIRST INST.       00166000 D
                                                        %                                         00167000 D
                                                        %                                         00168000 D
                                                        %                                         00169000 D
                                GETADDR:                %GET ADDRESS FIELD FROM COLUMNS N-M       00170000 D
                                                        %OF INPUT CARD. ADDRESS WILL BE IN        00171000 D
                                                        %DECIMAL. (N & M STAND FOR CARD COLS)     00172000 D
                                                        %                                         00173000 D
0052  4809 0003 0030 00F0            BEX                 % GET PRECEDING MEMORY WORD FROM SWI      00174000 D
0053  4809 0640 2030 00F0            AMPCR = A2          %SAVE RETURN                              00175000 D
0054  4809 0041 0B30 40FC            B L = B, CSAR                                                00176000 D
0055  0300 0003 0030 0090            COMP 24 = SAR;48=LIT  %ISOLATE LSB OF CARD(N-3)XN            00177000 D
```

```
0056  4809 0C40 CC30 00F0          0 R = A1                                    00178C00 D
0057  4809 A152 0C30 00FC          A1 EQL LIT                % TEST FOR BLANK ADDR FIELD       00179000 D
0058  5C0B 0000 0B30 00F0          IF TRUE THEN 0 = B; STEP ELSE SKIP  % IF BLANK ADDR TREAT AS 0 00180C00 D
0059  0150 0003 0030 0040          ALLBLANKS - 1 = MPCR      %OUTPUT EQUAL C IN B REG          00181C00 D
005A  0160 00C0 0C00 C06C          PACKED-1 = CPCR           %PACK DECIMAL NUMBERS INTO SINGLE WORD 00182C00 D
005B  0170 0000 0C00 0060          DECOCT-1 = CPCR           %DECODE  THE DECIMAL NUMBERS INTO OCTAL 00183C00 D
                              ALLBLANKS:                                       00184C00 D
005C  4809 C000 0C40 00FC          A2 = AMPCR                                  00185C00 D
005D  4809 0000 0C30 00FC          STEP                      % NEEDED FOR AMPCR ASSIGN PRIOR TO JUMP 00186C00 D
005E  482D 00C0 0030 00F0          JUMP                                        00187C00 D
                                                             %                00188C00 D
                              OUT:                           % THIS ROUTINE PERFORMS A MEMORY WRITE  00189C00 D
005F  9809 0000 0030 1CF0          MW2; IF SAI                                 00190C00 D
0060  9C08 C003 0030 00F0          WHEN SAI THEN 0; STEP                       00191C0C D
                                                             %                00192C00 D
                                                             %                00193C00 D
                              NEWCARD:                       %CALL IOP TO READ NEXT CARD THEN       00194C00 D
                                                             %JUMP TO CONTROL SPACE CARD ROUTINE.   00195C00 D
                                                             %THIS ROUTINE INSERTS SPACES INTO      00196C00 D
                                                             %HEXADDR AND CARDBUFFER                00197C00 D
                                                             %REMAINING 11 WORDS ARE FILLED WITH    00198CC0 D
                                                             %SPACES FOR ECHO PRINT ROUTINE.        00199000 D
0061  4809 00C0 0931 00F0          0 = BC8,LCTR                                00200C00 D
0062  01F0 0000 0020 00E0          31 = LIT                  %33 WORDS WILL HAVE SPACES        00201C0C D
0063  4809 20C3 001C 00F0          LIT = MAR2                                  00202C00 D
0064  0780 0003 0030 00A0          HEXADDR = LIT; COMP 16 =SAR                 00203C00 D
0065  4809 0C40 0C30 C0F0          B = MIR                   %PUT SPACES INTO MIR              00204C00 D
                                                             %                00205C0C D
                                                             %                00206C00 D
0066  0180 00C3 0030 0060    SPACES:  OUTPUT1 - 1 = CPCR     %WRITE SPACES INTO MEMORY         00207C00 D
0067  840B CF45 001E C0F0          IF NOT COV THEN BMAR + 1 = MAR2,INC; STEP ELSE SKIP 00208C00 D
0068  065C 0003 0030 0040          SPACES-1 = MPCR                             00209C00 D
                                                             %                00210C00 D
                                                             %                00211C00 D
                              CARDFETCH:                     % RETRIEVE CARD                 00212C0C D
0069  9809 20C0 0B30 00F0          LIT = B; IF SAI                             00213C0C D
006A  07AC 0000 0C00 00E0          CARDBUF = LIT             % CARDBUFF ADDRESS INTO B          00214000 D
006B  980E 0000 0030 00FC          IF IRQ THEN STEP ELSE SKIP  % IRQ INDICATES CRT INPUT       00215C00 D
006C  319C 00C3 0030 0060          CRTIN - 1 = CPCR                            00216C00 D
006D  4809 0C40 0C30 C0FC          B = MIR                   % CONTENTS FOR MAILBOX            00217C00 D
006E  01AC C0C0 0C3C 0060    BCRD:    EXTOP - 1 = CPCR                         00218C0C D
006F  06D0 00C3 C000 0040          BCRD - 1 = MPCR                             00219C0C D
0070  980B C0C0 0000 00F0          IF IRQ THEN STEP ELSE SKIP  % IRC -> FALSE THEN ECHO PRT (PTR) 00220000 D
0071  907C 0003 0030 C040          CONTROLCD - 1 = MPCR  % JUMP TO CONTROL CARD ROUTINE        00221C00 D
0072  4809 20C3 0030 00F0          LIT = MIR                                   00222000 D
0073  078C C0C0 0C0C 00EC          HEXADDR = LIT                               00223C0C D
0074  4809 00C1 0F3C 00F0          1 L = BBI                                   00224C0C D
0075  0600 0FC0 0030 0020          COMP 16 = SAR                               00225C0C D
0076  4809 0C40 C03C C0F0          B = MIR                   %MIR CONTAINS 1/HEXADDR           00226C00 D
                                                             %                00227C0C D
                                                             %                00228C00 D
                              ECHOPRINTCD:                   % ECHO CARD READ                00229C00 D
0077  0180 00C0 0CC0 0060          EXTOP - 1 = CPCR          % JUMP TO EXTOP ROUTINE           00230C0C D
0078  0760 C0C3 0C30 0C40          ECHOPRINTCD - 1 = MPCR % UNSUCCESSFUL IO OPERATION          00231C00 D
0079  007C 0003 0000 C040          CONTROLCD - 1 = MPCR  % JUMP TO CONTROL CARD ROUTINE        00232C00 D
                                                             %                00233C0C D
                                                             %                00234C00 D
                              EXTOP:                         %THIS ROUTINE PERFORMS THE IOP INTERFACE 00235000 D
                                                             %BY PASSING TO THE IOP THE FUNCTION    00236C00 D
                                                             %AND THE ADDRESS IN THE MAILBOX. THE IOP 00237C00 D
```

```
                                                                 *RETURNS THE STATUS OF THE
                                                                 *IN THE B REGISTER.
0C7A  1C08 0002 001C 00F0              SET GC2 WHEN GC2 THEN NOT 0 = MAR2
0C7B  9809 0000 0000 1CF0              MW2; IF SAI
0C7C  9F08 0000 0000 00F0              WHEN SAI THEN 0; SET INT
0C7D  D809 0000 0000 00F0              IF INT
0C7E  D808 0000 0000 00F0              WHEN INT THEN STEP
0C7F  A809 0000 0000 0CF0              BEX; MR2; IF RDC
0C80  AEC8 0C42 0000 00F0              WHEN RDC THEN NOT 0; RESET GC2
0C81  682F 0000 0000 00F0              IF ABT THEN JUMP ELSE RETN
                                                                 *
                                                                 *
                             OUTPUT1:
0C82  9809 0000 0000 1CF0              MW2; IF SAI
0C83  9C28 0000 0000 00F0              WHEN SAI THEN 0; JUMP
                                                                 *
                                                                 *
                                                                 *
0C84  4809 2001 2000 00F0    CRTIN:    LIT L = A2                * CRT READ CODE FOR CARDFETCH ROUTINE
0C85  0060 0000 0000 00AC              COMP 16 = SAR; 6 = LIT
0C86  4809 CC53 0800 00F0              A2 OR B = B
0C87  482D 0000 0000 00F0              JUMP
                                                                 *
                                                                 *
                             PACKED:                             * THIS ROUTINE MUST BE PASSED A 32 BIT
                                                                 * WORD IN UNPACKED DECIMAL FORM(4 DIGIT
                                                                 * DECIMAL NUMBER).  THE NUMBER IS PASSED
                                                                 * IN A1.  THE PACKED DECIMAL EQUIVALENT
                                                                 * OF THE NUMBER IS RETURNED IN B REGISTER
                                                                 * WHICH CONSISTS OF AN 8 DIGIT DECIMAL NO
0C88  49C9 0000 0000 00F0              SET LC1                   * THIS STEP IS OMITTED FOR A 4 DIGIT NO.
0C89  4809 0000 0030 00F0              0 = MIR                   * ZERO MIR
0C8A  4809 0000 0001 00F0    NEXT4:    LCTR                      * SET COUNTER TO PERFORM LOOP 4 TIMES
0C8B  3030 0000 0000 00B0              3 = LIT; COMP 4 = SAR
                             PACKTOP:
0C8C  4809 A001 4000 00F0              A1 L = A1                 * ELIMINATE HIGH ORDER 4 BITS OF MSB CHAR
0C8D  4809 A000 8F00 00F0              A1 R = BBI                * SHIFT A1 RIGHT 28 BITS AND "OR" WITH
                                                                 * MIR AND PLACE IN B REGISTER
0C8E  4809 0C41 0092 00F0              B L = MIR,INC             * LEFTMOST DIGIT NOW IN HIGH ORDER
                                                                 * POSITION OF MIR.
0C8F  940E A001 4000 00F0              IF NOT COV THEN A1 L = A1; STEP ELSE SKIP  * SHIFT NEXT DIGIT
                                                                 * INTO POS. FOR NEXT PASS THROUGH LOOP
0C90  088C 0000 0000 C040              PACKTOP - 1 = MPCR        *BRANCH TO TOP OF LOOP
0C91  2819 0000 0000 00FC              IF LC1 THEN SKIP          * CHECK TO SEE IF AN 8 DIGIT NUMBER
0C92  01C0 0000 0000 004C              PACKEND - 1 = MPCR        * JUMP TO THE END OF THE ROUTINE
                                                                 * IF ONLY A 4 DIGIT NUMBER WAS INPUT.
0C93  4809 0F45 001C 00F0              BMAR + 1 = MAR2           *PREPARE MAR TO READ NEXT 4 DIGITS
0C94  A809 0000 0000 0CF0              BEX; MR2; IF RDC          * B NOW CONTAINS LOW ORDER 4 DIGITS
                                                                 * OF AN 8 DIGIT NUMBER.
0C95  AC08 0C40 4000 00F0              WHEN RDC THEN B = A1      * WHEN READ COMPLETE, SET A1 EQUAL TO B
0C96  0890 0000 0000 0040              NEXT4 - 1 = MPCR          * BRANCH TO THE TOP OF THE LOOP
                             PACKEND:
0C97  482D 0000 0000 00F0              JUMP                      *RETURN TO CALLING ROUTINE WITH PACKED
                                                                 *NUMBER IN B REGISTER
                                                                 *
                                                                 *
                             DECOCT:                             * THIS ROUTINE WILL CONVERT DECIMAL
                                                                 * NUMBERS INTO THEIR OCTAL EQUIVALENTS.
                                                                 * THIS ROUTINE REQUIRES A PACKED DECIMAL
                                                                 * NUMBER PASSED VIA THE D REGISTER.  THE
```

```
                                                    % OCTAL EQUIVALENT WILL BE RETURNED IN B.  00298C00 D
0C98  4809 0C43 4C30 00FC          B = A1           % PACKED RESULTS ARRIVE IN B REG           00299C00 D
0C99  4809 0000 0B0C 0CFC          0 = B            % CLEAR B REGISTER                         00300C00 D
0C9A  4809 0000 0031 00F0          C = MIR, LCTR    % SET COUNTER FOR 7 ITERATIONS.            00301C00 D
0C9B  0060 00C3 0000 C0E0          6 = LIT                                                    00302C00 D
                                                    % PLACE DECIMAL NUMBER IN A1 AND CLEAR B. 00303C0C D
0C9C  4809 A000 9C30 C0F0  TOPDEC: A1 R = A3        % PUT HIGH ORDER 4 BITS OF A1 INTO         00304C0C D
0C9D  800C 0000 0C30 CC30          COMP 4 = SAR     % LOW ORDER 4 BITS OF A3.                 00305C00 D
0C9E  4809 A0C1 4C30 00F0          A1 L = A1        % PREPOSITION NEXT DECIMAL DIGIT IN A1.    00306C00 D
0C9F  48C9 EC41 1830 00F0          A3 + B L = A3, BAD % MULT DECIMAL DIGIT BY 8                00307C00 D
0CA0  900C 0000 0030 0030          COMP 3 = SAR                                               00308C00 D
0CA1  4809 CC41 0B32 C0F0          B L = B,INC      % MULT DECIMAL DIGIT BY 2, INCREMENT CTR   00309C00 D
0CA2  3000 00C0 0C30 C030          COMP 1 = SAR     %EFFECTIVELY MULT BY A FACTOR OF 10.       00310C00 D
0CA3  4809 EC43 0B30 00FC          A3 + B = B       % PARTIAL SUM OF CONVERSION                00311C00 D
0CA4  9C19 A000 903C 00F0          IF COV THEN A1 R = A3; SKIP  % GET LS DIGIT FROM A1 IF DONE  00312C00 D
0CA5  09B0 00C3 0030 0040          TOPDEC - 1 = MPCR  % GO TO TOP OF LOOP                      00313C0C D
0CA6  900C 0000 0030 003C          29 = SAR                                                   00314C00 D
0CA7  4809 EC40 0B30 00F0          A3 + B = B       % ADD LS DIGIT TO B AND RETURN OCTAL       00315C00 D
                                                    % VALUE IN B REGISTER.                     00316C00 D
0CA8  482D 00C3 0030 00F0          JUMP                                                       00317C00 D
                                                    %                                          00318C00 D
                                                    %                                          00319C0D D
                                DATA:               % CONVERT DECIMAL DATA IN COL 13-20 INTO 00320C00C D
                                                    % OCTAL AND STORE AT THE ADDRESS IN        00321C00 D
                                                    % COL 4-8                                  00322CC0 D
0CA9  051C 0000 0030 0060          GETADDR - 1 = CPCR % RETRIEVE ADDR IN COL 4-8 IN OCTAL      00323C00 D
0CAA  4809 0C40 1030 00F0          B = A3           % TRANSFER ADDR TO A3 REG                  00324C00 D
0CAB  4809 2CC3 001C 0DF0          LIT = MAR2       % LOAD ADDRESS OF COL 13-16                00325C00 D
0CAC  07D0 00D3 0030 C0E0          CARD13X16 = LIT                                            00326C00 D
0CAD  046C 00C3 0C30 0060          INPUT - 1 = CPCR % READ IN COL 13-16                       00327C00 D
0CAE  4809 0C43 4C30 D0F0          B = A1                                                     00328C0C D
0CAF  0870 00C3 0000 0060          PACKED - 1 = CPCR % PACK DATA IN COL 13-20                 00329C00 D
0CB0  4809 E0C0 2030 D0FC          A3 = A2          % SWITCH TEMP ADDRESS STORAGE             00330C00 D
0CB1  097C 0000 0000 0060          DECOCT - 1 = CPCR % CONVERT DECIMAL TO OCTAL               00331C00 D
0CB2  4809 0C40 0C30 00F0          B = MIR          % MIR CONTAINS DATA IN OCTAL               00332C00 D
0CB3  4809 C012 003C 00F0          A2 EQL 0         % SET UP LOGICAL TEST FOR BLANK ADDR       00333C00 D
0CB4  680B 00C0 003C 00F0          IF TRUE THEN STEP ELSE SKIP                                00334C00 D
0CB5  01DC 0000 0030 004D          INLINE - 1 = MPCR %IF ADDR = 0 ADD DATA IN NEXT ADDR       00335C0C D
00B6  4809 C0C3 001C 00F0          A2 = MAR2        % PUT MEMORY ADDR IN MAR2 FOR WRITE OP     00336C00 D
0CB7  05E0 00D3 0030 G040          OUT - 1 = MPCR   % WRITE DATA IN MEMORY AND GET NEXT CARD   00337C0C D
                                                    %                                          00338000 D
                                                    %                                          00339CC0 E
                                INLINE:             % THIS ROUTINE WILL INSERT DATA INTO THE  00340000 D
                                                    % ADDRESS CONTAINED IN THE NXTINSTR        00341C00 D
                                                    % REG.  THE DATA TO BE WRITTEN IS IN MIR.  00342C00 D
0CB8  4809 200D 201C C0F0          LIT = MAR2,A2    % PUT ADDR OF NEXTINST INTO MAR2, A2       00343C00 D
0CB9  0230 00D3 0030 C0E0          NXTINSTR = LIT                                             00344C00 D
0CBA  0460 00C0 0030 0D6D          INPUT - 1 = CPCR % READ CONTENTS OF NXTINSTR               00345C00 D
0CBB  4809 0C43 0C1C CDFD          B = MAR2         % PLACE CONTENTS OF NXTINSTR INTO MAR2     00346C00 D
0CBC  0810 00C3 DC0C 0060          OUTPUT1 - 1 = CPCR % PLACE DATA AT THAT ADDRESS(MAR2)       00347C00 D
0CBD  4809 0C46 0030 00F0          B + 1 = MIR      % CREATE NEXT ADDR FOR NXTINSTR            00348C00 D
0CBE  4809 C00D 0C1C C0F0          A2 = MAR2        % WRITE NEW ADDR IN NXTINSTR               00349C00 D
0CBF  05E0 00C3 0C30 0040          OUT - 1 = MPCR   % REG. AND GET THE NEXT CARD              00350C00 D
                                                    %                                          00351000 D
                                                    %                                          00352C00 D
                                SETINDEXREG:        % THIS ROUTINE WILL EXAMINE COL. 2 FOR    00353C00 D
                                                    % THE GENERAL REGISTER STACK NO. AND       00354000 D
                                                    % COL. 7-8 FOR THE REGISTER NO.  THE       00355G00 D
                                                    % CONTENTS OF COL. 13-20(ONLY 16-20 USED)  00356F00 D
                                                    % WILL THEN BE CONVERTED TO OCTAL AND      00357000 D
```

```
                                                    % STORED IN THE DESIRED GENERAL REGISTER.
0CC0  0460 0003 0030 C060           INPUT - 1 = CPCR     % REREAD COL. 1 - 4
0CC1  4809 0C41 CB30 00F0           B L = B              % ISOLATE COLUMN 2
0CC2  0000 00C0 0030 0030           COMP 8 = SAR
0CC3  4809 0C40 CB0C 00F0           B R = B,A1
0CC4  4809 A00E 0B30 00F0           A1 - 1 = B           % B IS NOW OCGEN REG. NO. 1) OR 1(NO. 2)
0CC5  4809 0C41 CB30 00F0           B L = B              % ADDRESS OF GENERAL REGISTER STACK
0CC6  8000 0000 0030 0030           COMP 4 = SAR         % MULTIPLY BY 16
0CC7  4809 0C41 0C28 00F0           B L = BR1,LMAR       % STORE BASE ADDR OF GEN REG STACK
0CC8  0000 00C3 0C00 0030           COMP 8 = SAR         % SHIFT TO PROPER POS. FOR BR1
0CC9  4809 200D 0C1C 00F0           LIT = MAR2           % LOAD ADDR OF CARD COL 5-8
0CCA  07B0 C0C3 0C00 C0EC           CARD5X8 = LIT
0CCB  0460 0003 0030 C060           INPUT - 1 = CPCR     % READ CARD COL. 5-8
0CCC  4809 0C41 CB3C 00FC           B L = B              % ISOLATE CARD COL 7-8
0CCD  0000 00C0 0030 0020           COMP 16 = SAR
0CCE  4809 0C43 CC00 C0F0           B R = A1             % A1 CONTAINS COL 7-8(GEN REG NO.)
0CCF  088C 0003 9030 C060           PACKED = CPCR        % JUMP TO 2ND LINE IN ROUTINE(4 DIGIT NO)
0CD0  0970 0000 0030 C060           DECOCT - 1 = CPCR    % B HAS OCTAL VALUE OF GEN REG NUMBER
0CD1  4809 0000 0C3C 20F0           ASR                  % SET MAR1 AS MOST RECENTLY REFERENCED
0CD2  4809 0F43 C00C 00F0           BMAR R = A1          % MOVE BASE ADDR TO A1 FROM BR1
0CD3  0C00 0000 0C3C 0010           8 = SAR              % SHIFT OMAR TO ISOLATE BR1
0CD4  4809 AC41 0C2C 00F0           A1 + B L = BR1       % CREATE GEN REG ADDR(BASE + OFFSET)
0CD5  07D0 00C0 0030 00B0           COMP 8 = SAR; CARD13X16 = LIT  % SHIFT FOR BR1
0CD6  4809 200D 001C 00F0           LIT = MAR2           % ADDRESS OF DATA
0CD7  0460 0003 0030 0060           INPUT - 1 = CPCR     % READ IN DATA
0CD8  0510 0003 0030 0060           GETADDR - 1 = CPCR
0CD9  4809 00D0 003C 20F0           ASR                  % SET MAR1 AS MOST RECENTLY REFERENCED
0CDA  4BC9 0F43 801C C0F0           BMAR R = MAR2        % LOAD GEN REG ADDR INTO MAR2
0CDB  0000 00C3 0C30 C010           8 = SAR              % SHIFT TO ISOLATE BR1
0CDC  4809 0C43 0030 00F0           B = MIR              % B HAS OCTAL VALUE OF DATA
0CDD  05EC 0000 0030 C040           OUT-1 = MPCR         %WRITE DATA FROM COL. 13-20 INTO
                                                         %SPECIFIED GENERAL REGISTER
                                                         %
                                                         %

                          BREAKPOINT:                    %SET UP BREAKPOINT REGISTER AND SET
                                                         % STATUS BITS
0CDE  4809 0000 0C0C 00F0           BEX                  % FILL B WITH CARD COL 1-4   (FROM SW1)
0CDF  4809 0C41 4C30 C0F0           B L = A1             % ISOLATE THE R,W,B CHARACTER
0CE0  0000 00C0 0030 0030           COMP 8 = SAR
0CE1  4809 A0C0 C030 00F0           A1 R = A1            %ISOLATE COLUMN 2
0CE2  0000 00C3 003C 003C           24 = SAR
0CE3  4809 200D 001C 00F0           LIT = MAR2           % WRITE PSW INTO B
0CE4  0200 0000 0030 00E0           PSW = LIT
0CE5  0460 00C0 0003 0060           INPUT - 1 = CPCR     % INPUT FROM MEMORY
0CE6  4809 A152 0C3C 00FC           A1 EQL LIT           %CHECK FOR CHARACTER :R:
0CE7  0290 00C0 0C30 00E0           41 = LIT
0CE8  6419 A152 0030 00F0           IF FALSE THEN A1 EQL LIT; SKIP
0CE9  01E0 00C0 0C00 C040           BRKREAD - 1 = MPCR   %JUMP TO  BREAKPOINT READ SUBFUNCTION
0CEA  036C 0000 0030 00E0           54 = LIT             %CHECK FOR :W: CHARACTER
0CEB  6019 C0C3 0000 00F0           IF FALSE THEN SKIP
0CEC  01F0 0000 0000 004C           BRKWRITE - 1 = MPCR  % JUMP TO BREAKPOINT WRITE SUBFUNCTION
0CED  020C 00C3 0030 0040           BRKBOTH -1 = MPCR    % DEFAULTS TO BREAKPOINT BOTH (R+W)
                                                         %
                                                         %

                          BRKREAD:                       % THIS ROUTINE SETS UP THE BREAKPOINT
                                                         % REGISTER AND SETS THE READ STATUS BITS
0CEE  4809 00C1 9030 00F0           1 L = A1
0CEF  8000 00C0 0C30 C010           COMP 20 = SAR
0CF0  4809 AC5C 005C 00F0           A1 OR B = MIR        % SET BIT # 2C OF PSW REG
0CF1  081C 00C3 0000 0060           OUTPUT1 - 1 = CPCR   % WRITE PSW INTO MEMORY
```

```
00F2  021C 0003 0030 0040          WRITEBRKPT - 1 = MPCR % JUMP TO WRITE BREAKPOINT REGISTER      00418C00 D
                                                         %                                       00419C00 D
                                   BRKWRITE:             % THIS ROUTINE SETS UP THE BREAKPOINT    00420C00 D
                                                         % REGISTER AND SETS THE WRITE STATUS BITS00421000 D
00F3  4809 0001 4030 00F0                1 L = A1                                                 00422C00 D
00F4  3000 0000 0000 0010                COMP 21 = SAR                                            00423000 D
00F5  4809 AC5C 0030 00F0                A1 OR B = MIR           %SET BIT 21 OF PSW REG           00424C00 D
00F6  0810 0003 003C 0060                OUTPUT1 - 1 = CPCR      % WRITE PSW INTO MEMORY          00425C0C E
00F7  022C 0003 0030 0040                WRITEBRKPT - 1 = MPCR % JUMP TO WRITE BREAKPOINT REGISTER 00426000 D
                                                         %                                       00427000 D
                                   BRKBOTH:              % THIS ROUTINE  SETS UP THE BREAKPOINT   00428C00 D
                                                         % REGISTER AND SETS THE STATUS BITS FOR  00429000 D
                                                         % BOTH READ AND WRITE                    00430C00 D
00F8  4809 2001 4030 00F0                LIT L = A1                                               00431C00 D
00F9  030C 0003 0000 00A0                COMP 16 = SAR; 48 = LIT  % SET UP A1 WITH MASK BIT POS. 21/20 00432000 D
00FA  4809 AC5C 0030 00F0                P OR A1 = MIR           % PUT BREAKPOINT CODE INTO SR1 BIT POS 00433000 D
                                                         % 21 AND 20                             00434C00 D
00FB  081C 00C0 0000 0060                OUTPUT1 - 1 = CPCR      % WRITE PSW INTO MEMORY          00435C00 D
                                                         %                                       00436C0C D
                                                         %                                       00437C0C D
                                   WRITEBRKPT:           %THIS ROUTINE WRITES THE DECIMAL ADDR    00438C00 D
                                                         % INTO THE BREAKPOINT REGISTER AND       00439C00 D
                                                         % AND INTO  REGISTER BR1                 00440C00 E
00FC  4809 2003 0C1C 00F0                LIT = MAR2              % RETRIEVE CARD1X4 CONTENTS       00441C00 D
00FD  07A0 0003 0C3C C0E0                CARD1X4 = LIT                                            00442C00 D
00FE  0460 00C0 0C30 0060                INPUT - 1 = CPCR        % INPUT INTO B                   00443C00 D
00FF  0510 0000 0000 C060                GETADDR - 1 = CPCR      % OBTAIN ADDRESS FROM COL 4-8    00444C00 D
0100  4809 2003 001C 00F0                LIT = MAR2                                               00445C00 D
0101  021C C0C0 C030 00E0               BKPT = LIT              % GET ADDRESS OF BKPT REG        00446C00 D
0102  4809 0C40 003C 00F0                B = MIR                 % B HAS BINARY  ADDRESS VALUE    00447C00 D
0103  4809 0C41 0C20 00F0                B L = BR1               %INSERT BREAKPOINT ADDR INTO BR1 00448C00 D
0104  0000 0000 0000 0030                COMP 8 = SAR                                             00449C00 D
0105  0810 0000 0000 C06C               OUTPUT1 - 1 = CPCR      % WRITE OUT ADDR INTO BKPT REG   00450C00 D
0106  060C 0C00 0000 0040                NEWCARD - 1 = MPCR      % RETURN TO NEW CARD ROUTINE     00451C00 D
                                                         %                                       00452C00 D
                                                         %                                       00453C00 D
                                   DOUBLEWORD:           %HANDLES SECOND WORD OF TWO WORD INSTR    00454C00 D
0107  4809 E000 880C 00FC                A3 R = B                % PUT COL 16 IN LSB               00455CC0 D
0108  07D0 0003 0000 00A0                16 = SAR; CARD13X16 = LIT                                00456C0C D
0109  4809 2003 001C 00F0                LIT = MAR2              %RELOAD ADDR OF CARD COL 13-16   00457C00 D
010A  052C 0003 0030 0060                GETADDR = CPCR          %OMIT PEX STEP IN GETADDR; B CONTAINS    00458C00 D
010B  4809 0C40 203C 00F0                B = A2                  % COL 16 DIGIT IN LSB OF B       00459C00 D
010C  023C 00C0 0030 C060                LOADINSTR - 1 = CPCR    %LOAD OCTAL VALUE OF DATA INTO SECOND    00460C00 D
010D  060C 0003 0030 0040                NEWCARD - 1 = MPCR      % WORD OF DOUBLEWORD INSTRUCTION AND     00461C00 D
                                                         %                                       00462CC0 D
                                                         %                                       00463C0C D
                                   INSTRUCTION:          % THIS ROUTINE WILL PACK AN INSTRUCTION   00464C00 D
                                                         % INPUT CARD INTO A 16 BIT INSTRUCTION   00465C00 D
                                                         % AND PLACE IT IN THE LS 16 BITS OF A    00466C0C D
                                                         % 32 BIT WORD.  RITYPE1 INSTRUCTIONS ARE 00467000 D
                                                         % ARE SELECTED  AND HANDLED BY A SEPARATE00468C00 D
                                                         % ROUTINE.                               00469C00 D
010E  4809 2003 0C1C 00F0                LIT = MAR2                                               00470C00 D
010F  37B0 00C3 0030 00A0                CARD5X8 = LIT; 21 = SAR                                  00471C00 D
0110  0460 C000 0000 0060                INPUT - 1 = CPCR        % READ THE CONTENTS OF COL. 5-8 INTO B    00472C00 D
0111  4809 0C40 A030 00F0                B R = A2                % ISOLATE 11 BITS (CHARACTER + 3 BITS)    00473C00 D
0112  4809 0C41 1030 00F0                B L = A3                % ISOLATE NEXT OCTAL DIGIT        00474C00 D
0113  3000 0000 003C C02C                COMP 13 = SAR                                            00475C00 D
0114  4809 E000 9030 00F0                A3 R = A3                                                00476C00 D
0115  9000 0003 0C30 0030                29 = SAR                                                 00477C00 D
```

```
0116  4809 C000 0090 00F0          A2 = AMPCR              % SET UP A Y-SELECT                    00478000 D
0117  4809 E641 0090 00F0          A3 + AMPCR L = AMPCR    % PACK OPCODE IN SIX BITS              00479000 D
0118  A000 C000 0030 0030          COMP 2 = SAR            % IN A2 AND SHIFT 2 BITS FOR NEXT FIELD 00480000 D
0119  4809 0C41 1030 40F0          B L = A3, CSAR          % ISOLATE FUNCTION BITS                 00481000 D
011A  27C0 C000 0000 C080          COMP 30 = SAR; CARD9X12 = LIT                                  00482000 D
011B  4809 E0C0 90D0 00F0          A3 R = A3                                                      00483000 D
011C  4809 E640 2030 00F0          A3 + AMPCR = A2         % PACK FUNCTION BITS (0-3) INTO A2      00484000 D
011D  4809 2003 001C 00F0          LIT = MAR2                                                     00485000 D
011E  0460 0C00 0090 0060          INPUT - 1 = CPCR        % READ IN NEXT 4 COL. (9-12)           00486000 D
011F  4809 2C56 9030 C0F0          B AND LIT = A1                                                 00487000 D
0120  0FFC 0000 0000 00E0          255 = LIT               % ISOLATE COL. 12 TO CHECK FOR COMMA    00488000 D
0121  4809 A152 0030 00FC          A1 EOL LIT                                                     00489000 D
0122  03A0 0000 0030 00E0          59 = LIT                % CODE FOR COMMA                        00490000 D
0123  5008 0000 0000 C0F0          IF FALSE THEN STEP ELSE SKIP % TREAT COL. 10-12 AS "0" FIELD   00491000 D
                                                          % OF RITYPE 1 INSTRUCTION              00492000 D
0124  024C 0003 0020 0040          RITYPE1 - 1 = MPCR      % TRANSFER TO RITYPE1 ROUTINE          00493000 D
                                                          % CARD HAS COL. 10-11 AS "A" FIELD     00494000 D
                                                          % COL. 13-14 AS "M" FIELD              00495000 D
0125  4809 0C41 0B30 C0F0          B L = B                 %ISOLATE CC 10,11                      00496000 D
0126  1000 0003 0030 0020          COMP 15 = SAR                                                  00497000 D
0127  4809 0C40 C030 C0F0          B R = A1                % ISOLATE 1 BIT OF CC 10 + 3 BITS       00498000 D
0128  8000 00C0 0030 0030          28 = SAR                                                       00499000 D
0129  4809 0C40 8B30 C0F0          B R = B                                                        00500000 D
012A  B070 0003 0000 00A0          23 = SAR; 7 = LIT       % ISOLATE CC 11 (3BITS) INTO B         00501000 D
012B  4809 2C56 0B30 00F0          B AND LIT = B                                                  00502000 D
012C  4809 AC40 0040 C0F0          A1 + B = AMPCR          % INSERT "A" FIELD INTO AMPCR          00503000 D
012D  4809 C0C1 2030 00F0          A2 L = A2                                                      00504000 D
012E  970C 0003 0000 00B0          COMP 4 = SAR; CARD13X16 = LIT                                  00505000 D
012F  4809 C641 2030 00F0          A2 + AMPCR L = A2       % 4 BITS FOR "M" FIELD                 00506000 D
0130  4809 2003 001C C0F0          LIT = MAR2                                                     00507000 D
0131  0460 0000 0030 0060          INPUT - 1 = CPCR                                               00508000 D
0132  4809 CC41 4B30 C0F0          B L = A1, B             %ISOLATE BIT WHICH IS COL 13 DIGIT(1,0) 00509000 D
0133  1C0C 0CC0 003C C030          COMP 7 = SAR                                                   00510000 D
0134  4809 A0C0 C030 00F0          A1 R = A1               %FILL A1 WITH EIT-0-0-0 (LSB)          00511000 D
0135  9000 00C0 0030 0030          28 = SAR                                                       00512000 D
0136  4809 0C41 1B30 00F0          B L = A3,B              %PUT COL14 IN LSB OF A3,B              00513000 D
0137  2000 C0C0 003C 0030          COMP 6 = SAR                                                   00514000 D
0138  4809 0C40 8B30 00F0          B R = B                 %PUT COL14 IN LSB OF B                 00515000 D
0139  7000 C0C0 0C30 0030          29 = SAR                                                       00516000 D
013A  4809 AC40 0E30 C0F0          A1 + B = B              % COMBINE COL. 13 AND 14                00517000 D
013B  4809 CC40 2000 C0F0          A2 + B = A2             % FINISH BUILDING 15 BIT INSTR          00518000 D
013C  0250 00C3 0000 006C          LOADINSTR - 1 = CPCR    % LOAD INSTRUCTION(A3 UNCHANGED)       00519000 D
013D  4809 E001 1000 C0F0          A3 L = A3               % A3 CONTAINS COLS. 15-16 IN UPPER HALF 00520000 D
013E  9000 00C0 C030 C030          COMP 3 = SAR                                                   00521000 D
013F  4809 E000 C030 C0F0          A3 R = A1               % SET UP COL. 15 IN LSB OF A1           00522000 D
0140  03A0 00C3 0C30 00B0          24 = SAR; 58 = LIT                                             00523000 D
0141  4809 A152 0C30 00FC          A1 EOL LIT              %CHECK FOR COMMA(,)                    00524000 D
0142  68CB 00C0 0000 00F0          IF TRUE THEN STEP ELSE SKIP                                    00525000 D
0143  106C 00C3 C03C 0040          DOUBLEWORD - 1 = MPCR   %COLS. 16-?- OF CARD CONTAIN DECIMAL   00526000 D
                                                          % "Y" FIELD                            00527000 D
0144  060C 00C0 0C3C 0C4C          NEWCARD - 1 = MPCR      % GET NEW CARD                         00528000 D
                                                          %                                      00529000 D
                                                          %                                      00530000 D
                                LOADINSTR:                % THIS ROUTINE WILL LOAD AN INSTRUCTION 00531000 D
                                                          % WHICH IS CONTAINED IN A2              00532000 D
                                                          % INTO THE NEXT APPROPRIATE S-MEMORY WORD00533000 D
0145  4809 0640 9030 C0F0          AMPCR = A1              % A1 IS A TEMP RETURN ADDRESS REG       00534000 D
0146  4809 20C3 001C 00F0          LIT = MAR2                                                     00535000 D
0147  0230 0000 C03C 00E0          NXTINSTR = LIT          % READ NEXT LOAD ADDRESS FROM NXTINSTR  00536000 D
0148  4809 00C0 0C31 00F0          LCTR                    % SAVE NXTINSTR IN CTR (COMPLEMENTED)   00537000 D
```

```
0149  0460 0000 0000 0060          INPUT - 1 = CPCR                                    00538000 D
014A  4809 0C40 001C 00F0          B = MAR2              % PLACE LOAD ADDRESS INTO MAR2 00539000 D
014B  4809 0C46 0030 00F0          B + 1 = B             % INCREMENT LOAD ADDRESS       00540000 D
014C  4809 C000 0090 00F0          A2 = MIR              % TRANSFER INSTRUCTION WORD IN A2 TO MIR 00541000 D
014D  0810 00C3 0030 0060          OUTPUT1 - 1 = CPCR    % AND WRITE INTO S-MEMORY      00542000 D
014E  4809 CB02 801C 00F0          NOT CTR R = MAR2      % WRITE LOAD ADDRESS           00543000 D
014F  3000 0000 0030 0030          29 = SAR              % INTO NXTINSTR                00544000 D
0150  4809 0C40 0090 00F0          B = MIR                                             00545000 D
0151  0810 00C3 0030 0060          OUTPUT1 - 1 = CPCR    % MEMORY WRITE                00546000 D
0152  4809 A0C3 003C 00F0          A1 = AMPCR            % REINSTALL RETURN ADDRESS     00547000 D
0153  4809 0000 0030 00F0          STEP                  % REQUIRED BEFORE JUMP         00548000 D
0154  4820 0003 0030 00F0          JUMP                                                00549000 D
                                                         %                             00550000 D
                                                         %                             00551000 D
                                MACHSTAT:                % THIS ROUTINE WILL DISPLAY    00552000 D
                                                         % THE AN/UYK-20 REG CONTENTS.  00553000 D
                                                         % IF M1  CONTROL CARD IS USED, THE 00554000 D
                                                         % OUTPUT WILL BE DISPLAYED ON THE CRT. 00555000 D
                                                         % THE SWI CONTAINS CC COL. 1-4. 00556000 D
0155  4809 2003 0C1C 00F0          LIT = MAR2, BEX       % LOAD ADDR OF SR#2 INTO MAR2  00557000 D
0156  0220 0000 0000 00E0          STATUS2 = LIT                                       00558000 D
0157  4809 0C40 8B30 00F0          B R = B               % CC COL. 2 INTO LS BYTE OF B  00559000 D
0158  03F0 0003 0030 00A0          16 = SAR; 63 = LIT                                  00560000 D
0159  4809 2C55 2030 C0F0          LIT AND B = A2        % MASK OFF COL. 2              00561000 D
015A  1000 0003 0030 0000          1 = SAR                                             00562000 D
015B  4809 C002 0030 00FC          A2 EQL 1              % CHECK FOR M1                 00563000 D
015C  5C19 18C1 C03C 00F0          IF TRUE THEN B10C C = A1; SKIP  % SET BIT 30 OF SR#2 FOR CRT 00564000 D
0150  4809 0000 4030 00F0          0 = A1                                              00565000 D
015E  0460 0003 0030 0060          INPUT - 1 = CPCR      % READ IN CONTENTS OF SR#2 INTO B 00566000 D
015F  4809 AC5C C030 00F0          A1 OR B = MIR         % SET APPROPRIATE SR#2 BITS    00567000 D
0160  0810 00C3 0030 0060          OUTPUT1 - 1 = CPCR    % WRITE NEW SR#2               00568000 D
0161  0260 0003 0030 0060          OUMPREG - 1 = CPCR    % DISPLAY REG CONTENTS         00569000 D
0162  0600 0000 0030 0040          NEWCARD - 1 = MPCR    % FETCH NEXT CARD              00570000 D
                                                         %                             00571000 D
                                                         %                             00572000 D
                                RESERVESPACE:            % THIS ROUTINE WILL RESERVE A DECIMAL 00573000 D
                                                         % NUMBER OF MEMORY CELLS AS INDICATED BY 00574000 D
                                                         % COL. 4-8.                    00575000 D
0163  0510 0003 0C30 0060          GETADDR - 1 = CPCR    % GET NUMBER OF CELLS AND CONVERT TO OCT. 00576000 D
0164  4809 0C40 4000 00F0          B = A1                % A1 CONTAINS NUMBER OF CELLS BEING SAVED 00577000 D
0165  4809 2003 001C 00F0          LIT = MAR2                                          00578000 D
0166  0230 0000 0000 C0E0          NXTINSTR = LIT                                      00579000 D
0167  0460 0003 0030 0060          INPUT - 1 = CPCR      % READ PRESENT NXTINSTR REG CONTENTS 00580000 D
0168  4809 AC40 0090 00F0          A1 + B = MIR          % NEXT INSTRUCTION ADDR TO MIR 00581000 D
0169  05E0 0003 0000 0040          OUT - 1 = MPCR        % WRITE OUT A NEW NEXT INST LOCATION 00582000 D
                                                         % INTO THE NXINSTR REG        00583000 D
                                                         %                             00584000 D
                                                         %                             00585000 D
                                RITYPE1:                 % THIS ROUTINE PACKS THE "D" FIELD INTO 00586000 D
                                                         % A2 (D FIELD IN CARD COL. 10-12) 00587000 D
016A  4809 0C40 9C00 00F0          B R = A3              % COL 10 TO LS 3 BITS OF A3    00588000 D
016B  0070 0003 0030 00A0          16 = SAR; 7 = LIT                                   00589000 D
016C  4809 E157 100C 00FC          A3 AND LIT L = A3     % ISOLATE COL 10 + 3 BITS INTO LSB OF A3 00590000 D
016D  3000 0003 0030 0030          COMP 3 = SAR                                        00591000 D
016E  4809 0C40 C000 00F0          B R = A1              % COL 11 TO LSB OF A1          00592000 D
016F  0000 00C0 C000 C010          B = SAR                                             00593000 D
0170  4809 A155 0040 00FC          A1 AND LIT = AMPCR     % ISOLATE COL 11 INTO AMPCR  00594000 D
0171  4809 E65D 1030 00F0          A3 OR AMPCR L = A3     % A3 = COLS. 10,11, + 3 BLANKS 00595000 D
0172  3000 0000 0030 003C          COMP 3 = SAR                                        00596000 D
0173  4809 2C56 0D30 00F0          LIT AND B = B          % ISOLATE COL. 12            00597000 D
```

```
0174  4809 EC50 0B0C 00FC          A3 OR B = B                 % B= CC00/CO=COL. 10,11,12          00598000 D
0175  4809 C001 2C00 00F0          A2 L = A2                   % PREPARE A2 FOR ADDITION OF B      00599000 D
0176  0000 00C3 0030 0030          COMP B = SAR                                                    00600000 D
0177  4809 CC40 2000 00F0          A2 + B = A2                 % PACK REMAINING BITS INTO A2        00601000 D
0178  1440 0000 0000 CC60          LOADINSTR - 1 = CPCR        % INSTRUCTION WORD RITYPE 1 COMPLETED 00602000 D
0179  05EC 0000 0C00 C040          OUT - 1 = MPCR                                                  00603000 D
                                                               %                                  00604000 D
                                                               %                                  00605000 D
                                SETPAGING:                     % THIS ROUTINE WILL SET THE PAGING BIT 00606000 D
                                                               % (BIT POS. 22) OF THE PSW          00607000 D
017A  4809 2003 001C 00F0          LIT = MAR2                                                      00608000 D
017B  A20C 0000 0000 00A0          PSW = LIT, 22 = SAR                                             00609000 D
017C  046C 0003 0000 0060          INPUT - 1 = CPCR            % READ IN CONTENTS OF PSW           00610000 D
017D  4809 0C41 8B30 40F0          B C = B, CSAR               % MOVE PAGING BIT TO LS BIT         00611000 D
017E  4809 2C53 8B00 00FC          LIT OR B C = B              % PAGING BIT SET(PAGING SELECTED)   00612000 D
017F  0010 0000 0000 C0E0          1 = LIT                                                         00613000 D
0180  4809 0C40 0090 C0F0          B = MIR                     % CONTENTS OF PSW INTO MIR          00614000 D
0181  05E0 C000 0C00 0040          OUT - 1 = MPCR                                                  00615000 D
                                                               %                                  00616000 D
                                                               %                                  00617000 D
                                CHARSTRING:                    % THIS ROUTINE WILL READ THE CHARACTER 00618000 D
                                                               % STRING STARTING IN COL. 9 UNTIL FINDING00619000 D
                                                               % A QUOTE (").  THE STRING WILL THEN BE 00620000 D
                                                               % STORED AT THE ADDRESS IN COL. 4-8.  IF 00621000 D
                                                               % COL. 4-8 IS BLANK, THE STORAGE ADDRESS 00622000 D
                                                               % WILL DEFAULT TO THE ADDRESS IN NXTINSTR00623000 D
0182  2809 0000 0000 00F0          IF LC1                      % CLEAR CONDITION BITS              00624000 D
0183  3809 0000 0C00 00F0          IF LC2                                                          00625000 D
0184  0510 0000 0030 0060          GETADDR - 1 = CPCR          % GET ADDRESS IN COL. 4-8           00626000 D
0185  4809 2003 001C 00F0          LIT = MAR2                  % MAR2 CONTAINS ADDR OF NEXT INSTRUCTION 00627000 D
0186  0230 0000 0C30 00E0          NXTINSTR = LIT                                                  00628000 D
0187  4809 0052 0C30 00F0          B EOL 0                     % CHECK FOR BLANK ADDRESS FIELD     00629000 D
0188  6A96 0000 0C00 C0F0          IF TRUE THEN SET LC2; STEP ELSE SKIP                            00630000 D
0189  0460 0000 0000 0060          INPUT - 1 = CPCR            % READ CONTENTS OF NXTINSTR         00631000 D
018A  4809 0C43 4C30 C0F0          B = A1                      % A1 CONTAINS ADDR TO BE WRITTEN INTO 00632000 D
018B  4809 2003 001C C0F0          LIT = MAR2                  % SET UP READ ADDR OF COL. 9-12 IN MAR2 00633000 D
018C  07D0 C000 0030 00EC          CARD9X12 = LIT                                                  00634000 D
018D  046C 0000 0C30 0060    MORECHAR:  INPUT - 1 = CPCR       % READ A 4 CHARACTER BLOCK          00635000 D
018E  0030 0003 0030 00EC          3 = LIT                                                         00636000 D
018F  4812 C003 0071 00F0          LCTR, SAVE                  % SET UP COUNTER FOR 4 ITERATIONS   00637000 D
0190  4809 0C41 8032 00F0          B C = B, MIR, INC           % SHIFT 1 CHARACTER INTO B TO BE ANALYZED00638000 D
0191  0000 0000 0000 0010          B = SAR                                                         00639000 D
0192  4809 2C56 2C00 00F0          LIT AND B = A2              % EXTRACT RIGHTMOST CHARACTER       00640000 D
0193  03F0 0003 0030 00E0          63 = LIT                    % SET UP 6 BIT MASK                 00641000 D
0194  4809 C152 0C30 00F0          A2 EOL LIT                                                      00642000 D
0195  00F0 0000 0C00 00E0          15=  LIT                    % TEST FOR " (0F)                   00643000 C
0196  6BC9 0003 0C00 00F0          IF TRUE THEN SET LC1        % IF A (") IS IN THE STRING SET LC1  00644000 D
0197  0210 00C0 0030 00E0          33 = LIT                    % ADD VALUE TO (") SO THAT SPACE APPEARS 00645000 D
0198  2FC9 2C40 0B30 C0F0          IF LC1 THEN LIT + B = B,MIR, SET LC1                            00646000 D
0199  8029 00C3 0C30 C0F0          IF NOT COV THEN JUMP        % CHECK IF LOOP DONE 4 TIMES        00647000 D
019A  4809 0F46 1030 C0F0          BMAR + 1 = A3               % SET UP NEXT READ ADDRESS          00648000 D
019B  4809 AC03 001C 00F0          A1 = MAR2                   % MAR2 NOW HAS WRITE ADDRESS        00649000 D
019C  081C 0003 0030 0060          OUTPUT1 - 1 = CPCR          % WRITE 4 CHAR BLOCK TO MEMORY      00650000 D
019D  4809 ACC0 4C00 00FC          A1 + 1 = A1                 % INCREMENT WRITE ADDRESS           00651000 D
019E  280B 00C0 0C00 00F0          IF LC1 THEN STEP ELSE SKIP  * CHECK IF (") WAS FOUND           00652000 D
019F  027C 0000 0000 0040          ENDCHARSTRING - 1 = MPCR                                        00653000 D
01A0  4809 EC03 001C 00F0          A3 = MAR2                   % A3 AND MAR2 HAVE READ ADDRESS     00654000 D
01A1  4809 E153 0C30 00F0          A3 LEO LIT                  % CHECK IF YOU ARE AT THE END OF THE CARD00655000 D
01A2  0800 00C3 003C 00E0          CARD77X8C = LIT                                                 00656000 D
01A3  700B 0C00 0000 00FC          IF TRUE THEN STEP ELSE SKIP                                     00657000 D
```

```
01A4  1800 0000 0000 0040          MORECHAR - 1 = MPCR     % READ IN MORE CHARACTERS          00659C00 D
01A5  3008 0000 CC30 00F0  ENDOCHARSTRING:  IF NOT LC2 THEN STEP ELSE SKIP                   00659C00 D
01A6  0600 0000 0000 0040          NEWCARD - 1 = MPCR      % GET NEXT CARD                    00660C00 D
01A7  4809 AC00 0030 00F0          A1 = MIR                % STORE A1 CONTENTS INTO MIR        00661C00 D
01A8  4809 2003 001C 00F0          LIT = MAR2              % LOAD MAR2 WITH ADDR OF NXTINSTR   00662C00 D
01A9  0230 0000 0030 00E0          NXTINSTR = LIT                                             00663C00 D
01AA  05EC 0000 0000 0040          OUT - 1 = MPCR          % UPDATE CONTENTS OF NXTINSTR       00664C00 D
                                                           %                                  00665C00 D
                                                           %                                  00666C00 D
                                   SWITCHES:               % THIS ROUTINE WILL EMULATE THE PHYSICAL 00667C00 D
                                                           % SETTING OF THE UYK-20 PROGRAM STOP    00668000 D
                                                           % SWITCHES 1 AND 2.  MAR2 CONTAINS ADDR 00669C00 D
                                                           % ADDRESS OF COL 1-9.  COL 2 WILL CONTAIN00670C00 D
                                                           % A 1(INDICATES SW#1 ON) OR A 2(INDICATES00671C00 D
                                                           % SW#2 IS ON).  BITS 30 AND 29 OF THE    00672C0C D
                                                           % PSW WILL BE SET TO A "1" TO INDICATE   00673C00 D
                                                           % SWITCH 1 AND SWITCH 2 RESPECTIVELY.    00674C00 D
01AB  0460 00C0 0030 C060          INPUT - 1 = CPCR        % REREAD COL. 1-9 INTO B            00675C00 D
01AC  4809 0C40 C00C C0FC          B R = A1                % COL. 2 INTO LS BYTE  OF A1        00676C00 D
01AD  03F0 0003 0000 00A0          16 = SAR, 63 = LIT                                         00677C00 D
01AE  4809 A156 4030 00F0          A1 AND LIT = A1         % MASK OUT LS BYTE OF A1           00678C00 D
01AF  4809 2003 001C 00F0          LIT = MAR2              % MAR2 CONTAINS ADDR OF PSW         00679000 D
01B0  0200 C0C0 0030 C0E0          PSW = LIT                                                  00680C00 D
01B1  0460 00C0 0030 C060          INPUT - 1 = CPCR        % READ INTO B THE CONTENTS OF THE PSW   00681C00 D
01B2  4809 A152 0030 00F0          A1 EQL LIT              % CHECK IF SW#1 SET                 00682C00 D
01B3  0010 0000 0030 00EC          1 = LIT                                                    00683C00 D
01B4  680B 0000 0030 00F0          IF TRUE THEN STEP ELSE SKIP                                00684C00 D
01B5  0280 0003 000C 0040          SETCOL30 - 1 = MPCR     % SET COL 30                       00685C0C D
01B6  4809 0C41 8B30 40F0          B C = B, CSAR           % SET COL 29(DEFAULTS TO SW#2 BEING SET) 00686C00 D
01B7  9010 0000 C030 00B0          29 = SAR, 1 = LIT                                          00687C00 D
01B8  4809 2C5C 0B3C 00F0          LIT OR B = B            % SET BIT 29                       00688C00 D
01B9  4809 CC41 8B3C 00F0          B C = B, MIR            % RESTORE B                        00689C00 D
01BA  05EC 00C0 C000 0040          OUT - 1 = MPCR                                             00690C00 D
01BB  4809 0C41 8B3C 40FC  SETCOL30:  B C = B, CSAR                                           00691C0C D
01BC  A010 0003 0000 00B0          30 = SAR, 1 = LIT                                          00692C00 D
01BD  4809 2C5C 0B3C 00F0          LIT OR B = B            % SET BIT 30                       00693C00 D
01BE  4809 CC41 8030 00F0          B C = MIR               % RESTORE B VALUE IN MIR           00694C00 D
01BF  05EC 00C0 003C 0040          OUT - 1 = MPCR                                             00695C00 D
                                                           %                                  00696C00 D
                                                           %                                  00697C00 D
01C0  4809 2003 0C1C 00F0          TRACE:  LIT = MAR2, BEX % THIS ROUTINE SETS THE 2 MS BITS IN SR#2 00698000 D
01C1  022C 00C0 0030 00E0          STATUS2 = LIT           % FOR DUMP REGISTER CALLS TO CRT OR PTR  00699C00 D
01C2  4809 0C40 8B3C 0CF0          B R = B                 % ISOLATE COL. 2 OF CONTROL CARD    00700C00 D
01C3  03F0 0003 C000 00A0          16 = SAR, 63 = LIT      % IF 1 THEN SET MS 2 BITS OF SR#2 (CRT)  00701C00 D
01C4  4809 2C56 2030 00F0          LIT AND B = A2          % A2 CONTAINS COL. 2 OF CONTROL CARD    00702C00 D
01C5  1000 C0C0 0030 C000          1 = SAR                                                    00703C00 D
01C6  4809 C0D2 0030 00F0          A2 EQL 1                % IS COL. 2 A 1?                    00704000 D
01C7  6C19 18C1 C030 C0FC          IF TRUE THEN B101 C = A1, SKIP                             00705CC0 D
01C8  4809 18C0 4030 00F0          B10D = A1               % PRINTER MASK                     00706000 D
01C9  0460 0003 0030 C060          INPUT - 1 = CPCR        % SR#2 INTO B                      00707C00 D
01CA  4809 AC5C 0030 00F0          A1 OR B = MIR           % SET MSB OF SR # 2                 00708C00 D
01CB  05E0 00C0 0000 0040          OUT - 1 = MPCR                                             00709C00 D
                                                           %                                  00710000 D
                                                           %                                  00711C00 D
                                   % * * * * * * * * * * * * AN/UYK-20 EMULATOR * * * * * * * * * * * * 00712C00 D
                                                           %                                  00713C00 D
                                                           %                                  00714C00 D
                                   % * * * * * * * * * * * * UTILITY ROUTINES * * * * * * * * * * * * 00715000 D
                                                           %                                  00716C00 D
                                                           %                                  00717C00 D
```

```
                                AEQM:                           % THIS ROUTINE TESTS IF THE "A" FIELD      00719000 D
                                                                % EQUALS THE "M" FIELD.  IF EQUAL LC2 SET  00720000 D
                                                                % A2 = RM; B = RA RETURNED                 00721000 D
01CC  4809 0000 0000 00F0               IF LC2                                                            00722000 D
01CD  4809 E000 8000 00F0               A3 R = B                                                          00723000 D
01CE  90FF 0003 0030 0080               4 = SAR; 15 = LIT                                                 00724000 D
01CF  4809 2056 0030 00FC               LIT AND B = B           % B CONTAINS "A" FIELD                    00725000 D
01D0  4809 E156 2000 00F0               A3 AND LIT = A2         % A2 CONTAINS "M" FIELD                   00726000 D
01D1  4809 CC52 0030 00F0               A2 EQL B                                                          00727000 D
01D2  5A49 0000 0000 00F0               IF TRUE THEN SET LC2                                              00728000 D
01D3  482D 0003 0030 00F0               JUMP                                                              00729000 D
                                                                %                                         00730000 D
                                                                %                                         00731000 D
01D4  4809 A001 2030 00F0       BUMP:   A1 L = A2               % THIS ROUTINE WILL INCREMENT THE PAR      00732000 D
01D5  0000 0000 0000 0020               COMP 16 = SAR           % TEST FOR WRAPAROUND AND CALL OPCODE      00733000 D
01D6  4809 C0F0 A000 C0F0               A2 R = A2               % PAR ISOLATED IN LHW OF A2                00734000 D
01D7  4809 C0C0 A000 00F0               A2 + 1 R = A2           % INCREMENT THE COPY OF THE PAR            00735000 D
01D8  4809 C013 0030 00F0               A2 GTR 0                % CHECK FOR WRAPAROUND                     00736000 D
01D9  74CB A0C0 4030 00F0               IF FALSE THEN A1 + 1 = A1 ELSE SKIP  % INCREMENT PAR BY 1         00737000 D
01DA  029C C000 0030 0040               OPCODE - 1 = MPCR       % CALL OPCODE FOR NEXT INSTRUCTION         00738000 D
01DB  4809 A0C0 C030 C0F0               A1 R = A1               % CLEAR OLD PAR                           00739000 D
01DC  0C00 0000 0030 0020               15 = SAR                                                          00740000 D
01DD  4809 A001 4000 00F0               A1 L = A1                                                         00741000 D
01DE  4809 0001 0B30 00F0               1 L = B                 % 1024 INTO B                             00742000 D
01DF  A0C0 0003 0030 0020               COMP 10 = SAR                                                     00743000 D
01E0  4809 AC50 4030 00F0               A1 OR B = A1            % SET PAR TO 1024                         00744000 D
01E1  02AC 00C0 0000 0040               OPCODE - 1 = MPCR       % FETCH NEXT INSTRUCTION                  00745000 D
                                                                %                                         00746000 D
                                                                %                                         00747000 D
                                BYTETEST:                       % THIS INSTRUCTION TEST THE LSB OF B       00748000 D
                                                                % WHICH CONTAINS EITHER (R(M)) OR (R(X))   00749000 D
                                                                % OR (R(M+1)) AND DIVIDES B BY TWO         00750000 D
                                                                % B CONTAINS RESULT                        00751000 D
                                                                % LC2 = 1 IMPLIES LS BYTE OF Y             00752000 D
                                                                % LC2 = 0 IMPLIES MS BYTE                  00753000 D
01E2  4809 0C40 0030 00F0               B                                                                 00754000 D
01E3  5A59 0003 0000 00F0               IF LST THEN SET LC2; SKIP  % IF LSB OF B EQUALS 1 THEN SET LC2    00755000 D
01E4  3809 0003 0030 00FC               IF LC2                  % ELSE CLEAR LC2                          00756000 D
01E5  4809 0C40 8B30 00F0               B R = B                 % B DIVIDED BY TWO                         00757000 D
01E6  1000 6000 0030 C000               1 = SAR                                                           00758000 D
01E7  482D 0003 0030 00F0               JUMP                                                              00759000 D
                                                                %                                         00760000 D
                                % * * * * * * * * * * * * * CARRY BIT SETTING * * * * * * * * * * * * *   00761000 D
                                                                %                                         00762000 D
01E8  280B 0000 0030 00F0       CARRY:  IF LC1 THEN STEP ELSE SKIP  % THIS ROUTINE WILL SET THE CARRY     00763000 D
01E9  0280 C000 0030 0040               SETCARRY - 1 = MPCR     % BIT IF LC1 IS SET UPON ENTRANCE          00764000 D
01EA  02C0 0000 0030 004C               CLEARC - 1 = MPCR       % OTHERWISE, CLEAR THE CARRY BIT           00765000 D
                                                                %                                         00766000 D
                                                                %                                         00767000 D
                                CLEARC:                         % THIS ROUTINE CLEARS THE CARRY BIT        00768000 D
                                                                % (POSITION 28) IN PSW                     00769000 D
01EB  4809 A001 C030 00FC               A1 C = A1                                                         00770000 D
01EC  8000 0003 0030 0030               28 = SAR                                                          00771000 D
01ED  4809 A0CE 4030 40F0               A1 AND B110 = A1,CSAR                                             00772000 D
01EE  4809 A001 0030 00F0               A1 C = A1                                                         00773000 D
01EF  482D 0003 0030 00F0               JUMP                                                              00774000 D
                                                                %                                         00775000 D
                                                                %                                         00776000 D
01F0  4809 A001 C030 00F0       SETCARRY:  A1 C = A1            % THIS ROUTINE SETS THE CARRY BIT IN       00777000 D
```

```
01F1  9000 0000 0030 0030          28 = SAR                 % BIT POS. 28 IN THE PSW      00778C00 0
01F2  4809 A00C 4030 40F0          A1 OR 1 = A1,CSAR                                     00779C00 0
01F3  4809 A0C1 C030 00F0          A1 C = A1                                             00780C00 0
01F4  4820 0003 000C C0F0          JUMP                                                  00781C00 0
                                                            %                            00782D00 0
% * * * * * * * * * * * * * * * END CARRY ROUTINES * * * * * * * * * * * * * * *      00783CC0 0
                                                            %                            00784C00 0
                                   CLEARBUFF:               %THIS ROUTINE INSERTS SPACES INTO THE  00785C00 0
                                                            %PRINT BUFFER (33 WORDS)      00786C00 0
01F5  4809 0640 2030 00F0            AMPCR = A2             % SAVE RETURN                00787000 0
01F6  4809 0000 0931 00FD            0 = BC8,LCTR                                        00788C00 0
01F7  01F0 0003 0000 00A0            COMP 16 = SAR,31 = LIT  % REQUIRED FOR BC8          00789C00 0
01F8  4809 2003 001C C0F0            LIT = MAR2                                          00790CD0 0
01F9  0990 0003 0030 00E0            PRINTBUFF = LIT                                     00791C00 0
01FA  4809 0C40 0030 00F0            B = MIR                %SPACES ARE WRITTEN INTO MIR 00792000 0
01FB  020C 0000 0000 0060  PBUFF:    EQUIPUT - 1 = CPCR                                  00793C00 0
01FC  8400 0F46 001E 00F0            IF NOT COV THEN PMAR + 1 = MAR2,INC; STEP ELSE SKIP 00794C00 0
01FD  1FAD 0000 0030 0040            PBUFF -1 = MPCR                                     00795C00 0
01FE  4809 C0C0 0030 00F0            A2 = AMPCR             % RESTORE RETURN             00796C00 0
01FF  4809 00C0 0030 00F0            STEP                   % DELAY                      00797000 0
0200  482D 0000 0030 C0FD            JUMP                                                00798C00 0
                                                            %                            00799C0C 0
                                                            %                            00800C00 0
% * * * * * * * * * * * * CONDITION CODE SETTING* * * * * * * * * * * * * * *      00801C00 0
                                                            %                            00802C00 0
                                                            %                            00803C00 0
                                   SETCC:                   % THIS ROUTINE WILL SET THE APPROPRIATE  00804D00 0
                                                            % CONDITION CODE IN BITS 24-25 OF THE    00805C00 0
                                                            % PSW.  THIS ROUTINE MUST BE PASSED      00806C00 0
                                                            % R(A) IN A2 AND R(M) OR (Y) IN B.       00807C00 0
0201  4809 CC52 0030 00F0            A2 EQL B               % CHECK IF R(A) = P(M) OR (Y) 00808C0C 0
0202  6419 CC4C 0030 00F0            IF FALSE THEN A2 XOR B; SKIP                        00809C00 0
0203  02E0 CC00 0000 0040            SET00 - 1 = MPCR       % SET CC CONDITION CODE      00810D00 0
0204  4819 00C0 0000 00F0            IF MST THEN SKIP       % UNLIKE SIGNS CASE          00811C00 0
0205  02FC 0000 0030 0040            LIKE - 1 = MPCR        % LIKE SIGNS CASE            00812C00 0
0206  4809 C0C0 0000 00F0            A2                     % A2 TO THE ADDER            00813CC0 0
0207  480B 0000 0C30 00FC            IF MST THEN STEP ELSE SKIP  % MST -> A2 WAS NEGATIVE 00814C00 0
0208  0300 0000 0C30 C040            SET11 - 1 = MPCR       % SET 11 CONDITION CODE      00815C00 0
0209  031C 00C0 C030 C040            SET01 - 1 = MPCR       % A2 POSITIVE, B NEGATIVE    00816000 0
020A  48C9 CC58 0030 C0FD  LIKE:     A2 GTR B               % LIKE SIGNS CASE            00817C0C 0
020B  70CB 0000 000C 00F0            IF TRUE THEN STEP ELSE SKIP                         00818C00 0
020C  0320 00C0 0C30 0040            SET01 - 1 = MPCR       % A2 > B                     00819C00 0
020D  0330 00C0 0C00 0040            SET11 - 1 = MPCR       % A2 < B                     00820C00 0
                                                            %                            00821CC0 0
                                                            %                            00822C00 0
                                   SETCCA:                  % THIS ROUTINE TESTS THE CONTENTS OF THE 00823C00 0
                                                            % DATA IN E WHICH IS THE RESULT OF AN    00824000 0
                                                            % ARITHMETIC INSTRUCTION                 00825C0D 0
                                                            % B CAN BE 16 OR 32 BITS LC1 SET = 32    00826C00 0
020E  2819 0003 0C0C 00F0            IF LC1 THEN SKIP       % TEST FOR DOUBLE WORD COMPARE 00827C00 0
020F  4809 CC41 0B30 00F0            B L = B                                             00828C00 0
0210  0C00 0000 0030 CC20            COMP 16 = SAR          % PREPARE SHIFT AMOUNT       00829C00 0
0211  4809 0C52 0030 C0F0            B EQL D                                             00830C0C 0
0212  6419 CC4D 0C30 00F0            IF FALSE THEN B; SKIP   % SKIP IF NON-ZERO          00831C00 0
0213  0340 0000 0030 0040            SET00 - 1 = MPCR                                    00832C00 0
0214  4819 00C0 0C30 00FC            IF MST THEN SKIP       % CHECK FOR NEGATIVE VALUE   00833C00 0
0215  0350 00C0 0000 0040            SET01 - 1 = MPCR                                    00834C00 0
0216  0360 C000 0000 C040            SET11 - 1 = MPCR                                    00835C00 0
                                                            %                            00836CC0 0
                                                            %                            00837C00 0
```

```
0217  4809 18C1 8B00 00F0      SET11:   B101 C = B          % DEFAULT TO (R(A)) < (R(M)) OR (Y)      00838C00 D
                                                            % ARITHMETIC TEST < 0 (NEG)             00839C00 D
0218  000C 0000 0000 001C               8 = SAR                                                    00840C00 D
0219  2809 00C0 0000 00F0               IF LC1               % JUST TO CLEAR FLAG SET IN SETCC      00841000 D
021A  4809 AC5C 4030 00F0               B OR A1 = A1                                               00842C00 D
021B  482D 0000 0030 00F0               JUMP                                                       00843C00 D
                                                            %                                       00844C00 D
                                                            %                                       00845C00 D
                                 SET00:                     % THIS ROUTINE WILL SET CONDITION CODE 0000846CC0 D
                                                            % IN BITS 24-25 OF THE PSW              00847C00 D
                                                            % ARITHMETIC TEST 0                     00848C00 D
021C  4809 18D9 8B30 0CFC               B010 C = B                                                 00849C00 D
021D  0000 0000 0C30 001C               8 = SAR                                                    00850C00 D
021E  4809 AC56 4030 C0F0               B AND A1 = A1                                              00851C00 D
021F  482D 0003 9030 C0F0               JUMP                                                       00852C00 D
                                 SET01:                     % THIS ROUTINE WILL SET CONDITION CODE 01C0853CC0 D
                                                            % IN BITS 24-25 OF THE PSW              00854C00 D
                                                            % ARITHMETIC TEST > 0 (POS)             00855C0C D
0220  4809 1800 8B30 C0F0               B100 R = B                                                 00856C0C D
0221  000C 0000 0000 001C               8 = SAR                                                    00857000 D
0222  4809 AC5C 4030 00FC               B OR A1 = A1                                               00858C00 D
0223  4809 1819 8B30 00FC               B011 C = B                                                 00859000 D
0224  3C00 0CC0 003C 0000               7 = SAR                                                    00860C00 D
0225  4809 AC56 4030 C0F0               B AND A1 = A1                                              00861C0C D
0226  482D 0003 0030 00F0               JUMP                                                       00862C00 D
                                                            %                                       00863C00 D
                                                            %                                       00864C00 D
                                 CHECKCC:                   % THIS ROUTINE WILL RETURN THE 2 COND  00865C0C D
                                                            % CODE BITS IN THE LS 2 BITS OF B       00866C0C D
0227  4809 A001 C030 40F0               A1 C = A1, CSAR      % SHIFT CC BITS TO LS 2 BIT POS.       00867C00 D
0228  9030 0003 0030 00A0               23 = SAR; 3 = LIT                                          00868000 D
0229  4809 A156 0B30 C0FC               A1 AND LIT = B       % ISOLATE CC BITS IN B                00869000 D
022A  4809 A001 CC30 00F0               A1 C = A1            % RESTORE A1                          00870C00 D
022B  482D 0003 0000 00F0               JUMP                                                       00871C0C D
                                                            %                                       00872000 D
                                 % * * * * * * * * * * END CONDITION CODE ROUTINES* * * * * * * * * * 00873C00 D
                                                            %                                       00874C00 D
                                 CONTENTSRA:                % THIS ROUTINE GETS (R(A)) IN B.  IT    00875C00 D
                                                            % ASSUMES THE INSTRUCTION IS IN B. THE  00876C00 D
                                                            % ADDRESS OF R(A) RETURNED IN MS WORD A3 00877000 D
022C  4809 0640 2000 C0F0               AMPCR = A2           % SAVE RETURN ADDRESS                  00878000 D
022D  4809 0C40 8B30 00F0               B R = B              % INSTRUCTION IS PASSED IN B           00879C00 D
022E  80F0 0000 000C 0080               4 = SAR; 15 = LIT                                          00880C00 D
022F  4809 2C56 0B30 00FC               LIT AND B = B        % ISOLATE THE "A" FIELD               00881C00 D
0230  0370 00C0 C03C 0060               REGSTACK - 1 = CFCR                                        00882C0C D
0231  4809 00C0 003C 24FC               ASE                                                        00883C00 D
0232  4809 0F40 0B3C C0F0               BMAR = B             % PUT ADDRESS OF R(A) INTO B           00884C00 D
0233  4809 0C41 0B30 00F0               B L = B              % MOVE ADDR OF R(A) TO MS WORD         00885CC0 D
0234  0000 0000 0C00 0020               COMP 16 = SAR                                              00886C00 D
0235  4809 EC5C 1030 C0FC               B OR A3 = A3         % UPPER WORD OF A3 HAS ADDR OF R(A)    00887C00 D
0236  0380 C003 0000 0060               EINPUT - 1 = CPCR    % (R(A)) IS IN B                      00888C00 D
0237  4809 C0F0 0C3C 00F0               A2 = AMPCR           % RESTORE RETURN ADDRESS               00889C00 D
0238  482D 0000 0030 00F0               JUMP                                                       00890CCC D
                                                            %                                       00891C00 D
                                                            %                                       00892000 D
                                 CONTENTSRM:                % THIS ROUTINE GETS (R(M)) IN B.  IT    00893C00 D
                                                            % ASSUMES THE INSTRUCTION IS IN A3.     00894000 D
0239  4809 0640 2C30 00FC               AMPCR = A2           % SAVE THE RETURN ADDRESS              00895C0C D
023A  4809 E156 0B3C 00F0               A3 AND LIT = B                                             00896CC0 D
023B  00FC 00C0 003C C0E0               15 = LIT                                                   00897C00 D
```

```
023C  0390 0003 0030 0060          REGSTACK - 1 = CPCR                                  00898C00 D
0230  03A0 C000 0070 0060          EINPUT - 1 = CPCR      % (R(M)) INTO B               00899000 D
023E  4809 C000 CC90 00F0          A2 = AMPCR                                           00900C00 D
023F  4809 0003 0000 00F0          STEP                                                 00901000 D
0240  4820 0000 0000 00F0          JUMP                                                 00902000 D
                                                    %                                   00903000 D
                                                    %                                   00904000 D
                                   DIV:             % THIS ROUTINE WILL DIVIDE A2/B, AND  00905000 D
                                                    % THE QUOTIENT IS RETURNED IN A3, REM IN 00906000 D
                                                    % A2. ALL NUMBERS MUST BE LEFT JUSTIFIED 00907000 D
                                                    % LC1 MUST BE SET FOR REG PAIR DIVIDEND  00908000 D
0241  4809 CC4C 0030 00F0          A2 XOR B                                             00909000 D
0242  4A49 0C40 0000 C0FC          B; IF MST THEN SET LC2  % NEG QUOTIENT FLAG           00910000 0
0243  4C09 0C5E 0B30 C0F0          IF MST THEN -B = B  % COMPLEMENT B                   00911000 D
0244  4809 0C43 8D30 00F0          B R = B                                              00912000 D
0245  0000 C000 0030 0020          16 = SAR                                             00913000 0
0246  4809 C0C0 0030 C0F0          A2                                                   00914000 0
0247  4C0E C0C2 2030 00F0          IF MST THEN NOT A2 = A2;STEP ELSE SKIP               00915000 0
0248  4809 C000 2030 C0F0          A2 + 1 = A2            % TWO'S COMPLEMENT            00916000 D
0249  2409 C000 A000 00F0          IF NOT LC1 THEN A2 R = A2  % RIGHT JUSTIFY FOR 16 BITS  00917000 0
024A  2809 0003 1000 00F0          0 = A3; IF LC1                                       00918000 D
024B  4809 0C52 0070 00F0          C EOL B                                              00919000 D
024C  680B 0FC0 0030 00F0          IF TRUE THEN STEP ELSE SKIP  % CHECK FOR ZERO DIVISOR  00920000 D
024D  0300 0C00 0030 0040          OVERFLOW - 1 = MPCR                                  00921000 D
024E  4809 0000 0031 24F0          LCTR; ASE              % SET UP CTR FOR 30 ITERATIONS  00922000 D
024F  01E0 0000 0000 00E0          30 = LIT                                             00923000 D
0250  4809 0003 1010 00F0          0 = A3, BR2            % CLEAR TRIAL DIVIDEND         00924000 D
0251  4809 0C43 001C 00F0          B = MAR2               % DIVISOR INTO BMAR            00925000 0
0252  4809 00C0 0B30 00F0          0 = B                  % CLEAR PARTIAL QUOTIENT       00926000 0
0253  4809 CF5E 0030 00F0          A2 LSS BMAR            % QUOTIENT = 0                 00927000 D
0254  740B 0000 1030 00F0          IF TRUE THEN 0 = A3; STEP ELSE SKIP                  00928000 0
0255  03C0 0003 0030 C040          ZQUO - 1 = MPCR                                      00929000 D
0256  4809 C001 2000 00F0          A2 L = A2              % SHIFT OFF MSB (ZERO)         00930000 D
0257  900C 00C0 0030 C030          COMP 1 = SAR                                         00931000 0
0258  4809 C0C0 0070 00F0  DIV1:'   A2                     % BRING DIVIDEND TO THE ADDER  00932000 D
0259  4C09 E00C 1030 00F0          IF MST THEN A3 OR 1 = A3  % BUILD TRIAL DIVIDEND     00933000 D
025A  4809 C0C1 2002 00F0          A2 L = A2, INC         % PREPARE NEXT MSB IN DIVIDEND  00934000 0
025B  3000 0000 0030 0030          COMP 1 = SAR                                         00935000 D
025C  4809 EF5E 0030 00F0          A3 GEQ BMAR            % CHECK IF TRIAL DIVIDEND >= DIVISOR  00936000 0
025D  780B 0003 0000 00F0          IF TRUE THEN STEP ELSE SKIP                          00937000 0
025E  0300 00C0 0000 004C          ONET - 1 = MPCR                                      00938000 D
025F  4809 0C41 0B30 00FC          B L = B                % PUT ZERO IN PARTIAL QUOTIENT  00939000 D
0260  900C 00C0 0C30 C030          COMP 1 = SAR                                         00940000 D
0261  03E0 C000 0000 0040          DIV2 - 1 = MPCR                                      00941000 D
0262  4809 EF5E 1030 00F0  ONET:   A3 - BMAR = A3         % TRIAL DIVIDEND - DIVISOR    00942000 0
0263  4809 0000 0070 00FC          STEP                                                 00943000 0
0264  4809 0C41 0B30 00FC          B L = B                                              00944000 D
0265  3010 00C0 0000 00B0          1 = LIT; COMP 1 = SAR                                00945000 D
0266  4809 2C5C 0B00 00FC          LIT OR B = B           % PUT 1 IN THE PARTIAL QUOTIENT  00946000 0
0267  4809 E001 1000 00F0  DIV2:   A3 L = A3              % PREPARE FOR NEXT DIGIT IN TRIAL DIVIDEN 00947000 D
0268  3000 0000 0C30 0030          COMP 1 = SAR                                         00948000 D
0269  900B 0003 0000 00F0          IF NOT COV THEN STEP ELSE SKIP                       00949000 0
026A  2570 0003 0030 0040          DIV1 - 1 = MPCR                                      00950000 D
026B  4809 E00C AC00 00F0          A3 R = A2              % REMAINDER INTO A2            00951000 D
026C  1000 0000 0030 0000          1 = SAR                                              00952000 D
026D  4809 0C40 1000 00F0          B = A3                 % QUOTIENT INTO A3             00953000 D
026E  4809 E00C 8B3C C0F0          A3 R = B               % EXAMINE UHW OF QUOTIENT      00954000 D
026F  000C 0000 0030 0020          16 = SAR                                             00955000 D
0270  4809 E001 1C00 00F0          A3 L = A3              % CLEAR UHW OF THE QUOTIENT    00956000 D
0271  4809 E00C 900C 00F0          A3 R = A3              % RIGHT JUSTIFY QUOTIENT       00957000 D
```

```
0273  4809 0C52 0030 00FC          0 EQL 0                    % CHECK FOR A 16 BIT QUOTIENT                    00959C00 D
0274  63C9 0000 0030 00F0          IF FALSE THEN SET LC1                                                       0096 0C0C D
0275  3C0C E002 1000 00FC          IF LC2 THEN NOT A3 = A3; STEP ELSE JUMP  % LC2 = NEG QUOT.                  00961C00 D
0276  4809 E0C3 1030 C0F0          A3 + 1 = A3                % COMPLEMENT QUOTIENT                             0C962C00 D
0277  4809 E0C1 1030 00F0          A3 L = A3                  % CLEAR UHW OF QUOTIENT                          0C963C00 D
0278  000C 0000 0000 0020          COMP 16 = SAR                                                               C0964C00 D
0279  4809 E0C0 9C30 00F0          A3 R = A3                  % RIGHT JUSTIFY QUOTIENT                         0C965C0C D
027A  4809 C0C2 2C00 C0F0          NOT A2 = A2                % COMPLEMENT REMAINDER                            CC966C00 D
027B  48C9 C0C1 2C30 00F0          A2 + 1 L = A2              % CLEAR UHW OF REMAINDER                         00967CC0 D
027C  482D C000 A000 C0FC          A2 R = A2; JUMP            % RIGHT JUSTIFY REMAINDER                        CC968C0C D
027D  3C0D C0C2 203C 00F0    ZQUO: IF LC2 THEN  NOT A2 = A2; STEP ELSE JUMP                                    0C969CC0 D
027E  482D C0C0 2C3C 00FC          A2 + 1 = A2; JUMP                                                           0C97CC00 D
                                                              %                                               0C971C00 D
                             % * * * * * * * * * * * * * * * * DEBUGGER * * * * * * ** * * * * * * * * * * * 0C972C0C E
                                                              %                                               00973C00 D
                             DUMPREG:                         % THIS ROUTINE WILL DUMP OUT THE                 0C974C00 E
                                                              % EMULATOR:S MAIN MEMORY MAPPING THROUGH         00975C00 D
                                                              % THE RTC ADDR AND THE FOLLOWING                 0C976C00 D
                                                              % D-MACHINE REGISTERS:  AMPCR, A1, A2,           0C977C00 D
                                                              % A3, MIR, AND BR1                               00978C00 D
027F  4809 0000 0D3C 00F0          DMI                        % SAVE MIR IN B REGISTER                         0C979C00 D
0280  4809 0640 0030 00FC          AMPCR = MIR                % SAVE AMPCR                                     0098CC00 D
0281  4809 20D3 001C C0F0          LIT = MAR2                                                                  0C981C0C D
0282  026C 0003 0C00 F0EC          WORKSPACE = LIT                                                             C098 2C00 D
0283  03F0 00C0 000C 0060          EOUTPUT - 1 = CPCR         % WRITE AMPCR INTO WORKSPACE                     0C983C00 D
0284  4809 0F45 0C1C C0F0          BMAR + 1 = MAR2                                                             0C984C00 D
0285  4809 ACC0 003C 00F0          A1 = MIR                                                                    0C985C0C D
0286  040C 00C0 0C0C 006C          EOUTPUT - 1 = CPCR         % WRITE A1 INTO WORKSPACE + 1                    00986C00 D
0287  4809 0F45 0C1C C0F0          BMAR + 1 = MAR2                                                             0C987C00 D
0288  4009 CC03 0030 00FC          A2 = MIR                                                                    C098 8C00 D
0289  041C 0000 0C00 0060          EOUTPUT - 1 = CPCR         % WRITE A2 INTO WORKSPACE + 2                    00989C0C D
028A  4809 0F46 001C 00F0          BMAR + 1 = MAR2                                                             0C99CC00 D
028B  4809 E0C3 0C30 00F0          A3 = MIR                                                                    00991000 D
028C  0420 0000 0030 0060          EOUTPUT - 1 = CPCR         % WRITE A3 INTO WORKSPACE + 3                    00992C00 D
028D  4809 0F45 001C C0FC          BMAR + 1 = MAR2                                                             0C993C00 D
028E  4809 0C40 0030 00F0          B = MIR                                                                     0C994C00 D
028F  0430 00C0 0000 0060          EOUTPUT - 1 = CPCR         % WRITE MIR INTO WORKSPACE + 4                   0C995C00 D
0290  4809 0F45 001C C0FC          BMAR + 1 = MAR2                                                             0C996C00 D
0291  4809 00C0 0030 20FC          ASR                        % REFERENCE BR1                                  00997000 D
0292  48C9 0F40 0C30 C0F0          BMAR R = MIR                                                                0C998C00 D
0293  0000 0000 0030 C010          B = SAR                                                                     0C999C00 D
0294  0440 0000 0000 C060          EOUTPUT - 1 = CPCR         % WRITE BR1 INTO WORKSPACE + 5                   010CC000 D
0295  4809 20C0 103C C0FC          LIT = A3                   % A3 HOLDS READ ADDR                             01001C00 D
0296  000C 0000 003C 00B0          GREG1 = LIT; COMP B = SAR                                                   0100 2C0C D
0297  4809 2001 0C20 00FC          LIT L = BR1                % USE FR1 AS HOLDER OF READ ADDRESS               01003C00 D
0298  1F4C 00C0 003C C060    NEXTEIGHT:  CLEARBUFF - 1 = CPCR  % CLEAR PRINTBUFF                               0100 4C00 D
0299  4809 EC03 0C1C C0F0          A3 = MAR2                  % SET UP READ ADDRESS                            01005C00 D
029A  4809 20C9 4030 00F0          LIT = A1                   % PRINTBUFF ADDR TO BE WRITTEN INTO              01006C0C D
029B  099C 0003 0000 00E0          PRINTBUFF = LIT                                                             01007C00 D
029C  0450 0000 0000 006C    NEXTREG: EINPUT - 1 = CPCR       % READ THE CONTENTS OF A GENERAL REG             01008C0C D
                                                              % INTO THE B REGISTER                            01009C00 D
029D  4809 0F41 002C C0F0          BMAR L = BR1                                                                01010CCC D
029E  00C0 0CC0 0000 0030          COMP 8 = SAR                                                                0101 1C0C D
029F  4809 0C41 9031 C0F0          B C = A3, LCTR             % PUT B INTO A3, INTERCHANGE MS 16 BITS           01012C00 D
02A0  0070 00C3 0C00 C0A0          16 = SAR; 7 = LIT          % AND LS 16 BITS                                 01013C00 D
02A1  4809 0003 0B3C C0F0          0 = MIR,B                  % CLEAR MIR AND B                                01014C00 D
02A2  2879 E155 2000 00F0    NEXTDIGIT:  A3 AND LIT = A2; IF LC1  % A2 CONTAINS LS 4 BITS OF A REG           01015C00 D
02A3  90F0 0000 0000 0080          15 = LIT ; 4 = SAR                                                          C1016 0C0 D
02A4  48C9 E000 900C C0F0          A3 R = A3                  % SHIFT A3 RIGHT TO POSITION NEXT 4 BITS         C1017C0C D
```

```
02A5  4809 C153 0030 00FC    A2 GTR LIT                 % CHECK IF 4 DIGIT HEX VALUE IS A-F   01018000 D
02A6  0090 0009 0030 00E0    9 = LIT                                                          01019000 D
02A7  7C0B C140 2C30 00F0    IF TRUE THEN A2 + LIT = A2; STEP ELSE SKIP  % ADD 7 FOR BCL      01020000 D
02A8  0070 0009 0C00 00E0    7 = LIT                                                          01021000 D
02A9  4809 CC41 8B30 00F0    A2 + B C = B               % PLACE 4 DIGIT VALUE INTO BC HEX WORD) 01022000 D
02AA  0000 0000 0030 0010    B = SAR                                                          01023000 D
02AB  4809 0C40 0090 00F0    B = MIR                                                          01024000 D
02AC  4809 0B02 8B02 C0F0    NOT CTR R = B, INC     % USE B TO CHECK COUNTER VALUE            01025000 D
02AD  0040 00C0 0030 00B0    24 = SAR; 4 = LIT                                                01026000 D
02AE  4809 A003 001C 00F0    A1 = MAR2              % PROPER WRITE ADDRESS                     01027000 D
02AF  4809 2C52 0C30 C0F0    LIT EOL B              % TEST COUNTER FOR 4 ITERATIONS            01028000 D
02B0  6BCB 0003 0C30 C0F0    IF TRUE THEN SET LC1; STEP ELSE SKIP  % TRUE MEANS HALFWORD      01029000 D
02B1  0460 00C0 0030 C060    EOUTPUT - 1 = CPCR     % COMPLETED-WRITE IT INTO MEMORY          01030000 D
02B2  2FC9 A0C0 9C30 00F0    IF LC1 THEN SET LC1; A1 + 1 = A1 % INCREMENT PRINTBUFF ADDR      01031000 D
02B3  2FC9 0000 0B30 00F0    IF LC1 THEN SET LC1; 0 = B  % CLEAR B FOR NEXT HALFWORD           01032000 D
02B4  280B 0000 003C 00F0    IF LC1 THEN STEP ELSE SKIP  % 16 BITS OF THE GIVEN REGISTER       01033000 D
02B5  2A1C 00C0 0000 004C    NEXTOIGIT - 1 = MPCR  %HAVE BEEN CONVERTED TO BCL FORMAT          01034000 D
0286  4809 0000 C030 00F0    BMI                    % RESTORE PARTIAL B CONSTRUCT             01035000 D
02B7  800B 0000 C00C C0F0    IF NOT COV THEN STEP ELSE SKIP                                   01036000 D
02B8  2A1C 00C0 0000 0040    NEXTDIGIT - 1 = MPCR   % LOOP DONE 8 TIMES FOR 32 BITS OF EACH   01037000 D
                                                    % GENERAL REGISTER                        01038000 D
02B9  4809 A003 001C 00F0    A1 = MAR2              % PREPARE TO WRITE OUT LAST HALFWORD       01039000 D
02BA  0470 00C3 0030 0060    EOUTPUT - 1 = CPCR     % AT THIS POINT, A FULL REG IS DONE        01040000 D
02BB  4809 00C0 0030 20FC    ASR                    % REFERENCE BR1                           01041000 D
02BC  4809 0F40 9000 00F0    BMAR R = A3            % RESTORE OLD READ ADDRESS IN A3           01042000 D
0280  000C 0000 003C C010    8 = SAR                                                          01043000 E
02BE  4809 E0C3 1030 00F0    A3 + 1 = A3            % SET UP NEXT READ ADDRESS                 01044000 D
02BF  4809 E152 0030 C0F0    A3 EOL LIT                                                       01045000 D
02C0  008C 0003 0030 00E0    8 = LIT                                                          01046000 D
02C1  6BCB E152 0C30 00F0    A3 EOL LIT; IF TRUE THEN SET LC1;STEP ELSE SKIP                  01047000 D
02C2  0480 0003 003C 0060    WRITEBUFF - 1 = CPCR                                             01048000 D
02C3  010C 0003 003C 00E0    16 = LIT                                                         01049000 D
02C4  6BCB E152 0C30 00F0    A3 EOL LIT; IF TRUE THEN SET LC1;STEP ELSE SKIP                  01050000 D
02C5  0490 0000 0000 0060    WRITEBUFF - 1 = CPCR                                             01051000 D
02C6  0180 00C3 0000 C0E0    24 = LIT                                                         01052000 D
02C7  6BCB E152 0C00 C0F0    A3 EOL LIT; IF TRUE THEN SET LC1;STEP ELSE SKIP                  01053000 D
02C8  04A0 0003 003C 0060    WRITEBUFF - 1 = CPCR                                             01054000 D
02C9  020C 0003 0030 00E0    32 = LIT                                                         01055000 D
02CA  6BCB E152 0000 00F0    A3 EOL LIT; IF TRUE THEN STEP; SET LC1 ELSE SKIP                 01056000 D
02CB  04BC 0000 0C30 C060    WRITEBUFF - 1 = CPCR                                             01057000 D
02CC  029C 00C3 0030 C0E0    40 = LIT                                                         01058000 D
02CD  6BCB E152 0030 C0F0    A3 EOL LIT; IF TRUE THEN STEP; SET LC1 ELSE SKIP                 01059000 D
02CE  04C0 0003 0C30 0060    WRITEBUFF - 1 = CPCR                                             01060000 D
02CF  4809 E003 001C 00F0    A3 = MAR2              % MAR2 CONTAINS ADDRESS OF NEXT REG.      01061000 D
0200  280B C000 0030 00F0    IF LC1 THEN STEP ELSE SKIP  % ARE ARE WE FINISHED WITH ONE LINE  01062000 D
0201  297C 00C3 0030 C04C    NEXTEIGHT - 1 = MPCR                                             01063000 D
0202  4809 A140 903C 00F0    A1 + LIT = A1          % PUT SPACE BETWEEN REGISTER VALUES        01064000 D
02D3  0020 0000 003C 00E0    2 = LIT                                                          01065000 D
0204  4809 E158 0030 00F0    A3 LEO LIT             % CHECK IF ALL REG HAVE BEEN DUMPED        01066000 D
0205  02F0 0003 0C00 00E0    47 = LIT               % CURRENTLY, ROUTINE IS SET FOR 48 REGS    01067000 D
0206  700B 0003 0000 00F0    IF TRUE THEN STEP ELSE SKIP                                      01068000 D
02D7  29BC 00C3 003C 0040    NEXTREG - 1 = MPCR     % READ ANOTHER REGISTER                   01069000 D
0208  040C 0000 0C30 0060    WRITEBUFF - 1 = CPCR   % PRINT OUT LAST 8 REGISTERS              01070000 D
02D9  1F4C 00C0 0030 0060    CLEARBUFF - 1 = CPCR   % CLEAR THE PRINT BUFFER                  01071000 D
020A  04EC 00C3 0CD0 0060    WRITEBUFF - 1 = CPCR   % JUST TO CREATE A BLANK LINE             01072000 D
020B  04F0 00C0 0030 C060    WRITEBUFF - 1 = CPCR   % JUST TO CREATE A BLANK LINE             01073000 D
02DC  050C 0003 003C C060    WRITEBUFF - 1 = CPCR   % JUST TO CREATE A BLANK LINE             01074000 D
020D  0510 0000 0C30 0060    WRITEBUFF - 1 = CPCR   % JUST TO CREATE A BLANK LINE             01075000 D
02DE  4809 20C3 001C 00FC    LIT + 1 = MAR2         % LOAD THE ADDRESS OF A1 INTO MAR2         01076000 D
02DF  0260 0000 0030 C0E0    WORKSPACE = LIT                                                  01077000 D
```

```
02E0  052C 0000 0000 0060          EINPUT - 1 = CPCR      % READ IN CONTENTS OF A1                01078C00 D
02E1  4809 0C40 9000 C0F0          B = A1                 % RESTORE A1                             01079C00 D
02E2  4809 0F46 C01C 00F0          BMAR + 1 = MAR2        % CONSTRUCT ADDRESS OF A2                0108 0C00 D
02E3  0530 0000 0030 0060          EINPUT - 1 = CPCR      % READ IN A2                            01081C00 D
02E4  4809 0C40 2030 C0F0          B = A2                 % RESTORE A2                             01082CC0 D
02E5  4809 0F46 001C C0F0          BMAR + 1 = MAR2        % CONSTRUCT ADDRESS OF A3                0108 300C D
02E6  0540 0CC0 0030 006C          EINPUT - 1 = CPCR      % READ IN A3                            01084CC0 D
02E7  4809 CC40 1C30 00FC          B = A3                 % RESTORE A3                             01085C00 D
02E8  4809 0F46 0C1C 00F6          BMAR + 1 = MAR2        % CREATE ADDR OF MIR(WORKSPACE + 4)      01086C00 D
02E9  0550 00C0 0030 0060          EINPUT - 1 = CPCR      % READ IN MIR                           01087C00 D
02EA  4809 0C40 003C 00F0          B = MIR                % RESTORE MIR                            01088C00 D
02EB  4809 0F46 001C 00F0          BMAR + 1 = MAR2        % CREATE ADDR OF BR1                     01089CC0 D
02EC  0560 00C0 0000 0060          EINPUT - 1 = CPCR                                              0109 0C00 D
02ED  4809 0C4L 0020 00F0          B L = BR1              % RESTORE ER1                           01091C00 0
02EE  0260 0C00 0000 G0B0          COMP B = SAR ; WORKSPACE = LIT                                 01092C00 0
02EF  4809 2003 001C 00F0          LIT = MAR2             % PUT ADDRESS OF AMPCR IN MAR2           01093C00 D
02F0  0570 0003 0070 0060          EINPUT - 1 = CPCR      % READ IN OLD AMPCR                     01094C00 D
02F1  4809 0C40 0040 00F0          B = AMPCR              % RESTORE OLD AMPCR                      01095C00 D
02F2  4809 0000 0000 00F0          STEP                                                           01096C00 D
02F3  4820 00C0 000C 00FC          JUMP                                                           01097C0C 0
                                                          %                                       01098C00 D
                                                          %                                       01099000 D
                                   EINPUT:                % THIS ROUTINE READS THE CONTENT OF THE  01100C00 D
                                                          % MEMORY ADDRESS IN MAR2                 011C1C00 D
02F4  A809 00C3 003C 0CF0          MR2; IF ROC            % PERFORM READ                           C11C2C00 D
02F5  AC28 00C0 0C0C 00F0          WHEN ROC THEN BEX; JUMP   % READ CONTENTS OF MAR2 INTO B        01103C00 D
                                                          %                                       C1104C00 D
                                                          %                                       01105000 D
                                   EOUTPUT:               % THIS ROUTINE WRITES MIR INTO THE       01106C00 D
                                                          % MEMORY ADDRESS OF MAR2                 01107C00 D
02F6  9809 0000 0C00 1CF0          MW2; IF SAI            % MEMORY WRITE                           01108C00 D
02F7  9C28 0000 000C 00F0          WHEN SAI THEN 0; JUMP  % WHEN WRITE COMPLETED JUMP             C11C9C00 D
                                                          %                                       0111C000 D
                                                          %                                       01111C00 D
                                   EXTIO:                 % THIS ROUTINE DOES EXTERNAL I/O         01112C00 D
                                                          % INTERFACE BETWEEN THE UYK20 EMULATION  01113C0C D
                                                          % AND THE IOP.  B CONTAINS THE CONTENTS  01114C00 D
                                                          % OF MAILBOX - 1  IF REQUIRED.           01115C00 D
02F8  1C88 0013 0C1C 00F0          SET GC2; WHEN GC2 THEN B111=MAR2                               01116C00 D
02F9  9809 00D3 001C 1CF0          MW2; B110=MAR2; IF SAI                                         01117C00 D
02FA  9C08 CC40 003C C0F0          WHEN SAI THEN B = MIR                                          01118C00 D
02FB  4809 C013 001C 1CF0          B111 = MAR2; MW2       % WRITE B INTO MAILBOX - 1               01119C00 D
02FC  9B02 0C00 0C30 C0F0          WHEN SAI THEN SET INT                                          0112C000 D
02FD  D008 00C3 CC3C C0FC          WHEN NOT INT THEN STEP  \                                      01121C00 D
02FE  3EC8 C000 0C3C 0CF0          WHEN INT THEN BEX; MR2                                         01122C00 D
02FF  AEC8 0C42 0C30 00F0          WHEN ROC THEN NOT B; RESET GC2                                 C1123C0C 0
0300  682F 00C0 C03C 00F0          IF ABT THEN JUMP ELSE RETN                                     01124C00 D
                                                          %                                       01125C00 D
                                                          %                                       C1126C00 D
                                   FAULT:                 % THIS ROUTINE PRINTS OUT THE ERROR      C1127C00 D
                                                          % MESSAGE "FAULT INTERRUPT"             01128C00 D
                                                          % INSERTED BY LOADER JCL AND STOPS THE  C1129C00 D
                                                          % MACHINE                               01130C00 D
0301  0580 0000 0000 006C             SAVEREG - 1 = CPCR                                          01131C00 D
0302  4809 0641 0C20 C0F0              AMPCR L = BR1                                               01132C00 D
0303  0CE0 0000 C00C C0C0              ERRORLIST = AMPCR      % CREATE ERRORLIST ADDRESS IN BR1     01133C00 D
0304  00DC 00D3 000C 0030              COMP B = SAR                                                C1134C0C D
0305  0590 0CC0 003C C040              PRTERR - 1 = MPCR                                           01135C00 D
                                                          %                                       01136C00 D
                                                          %                                       01137CC0 0
```

```
                                                % * * * * * * * * INPUT OUTPUT CONTROLLER * * * * * * * * * * * * * * * * * * * * 01138C00 D
                                                %                                                                                 01139C00 D
                                    COMPRES:    %  THIS ROUTINE COMPRESSES /N/UYK20                                             01140C00 D
                                                %  16 BIT BUFFERS (LHW OF 32 BIT BURROUGHS                                       01141C00 D
                                                %  S-MEMORY WORD) INTO A 32 BIT WORD                                             01142C00 D
                                                %  FORMAT COMPATIBLE WITH THE IOP                                                01143000 D
                                                %  UPON ENTRY, BR1 CONTAINS BUFFER ADDRESS                                       01144C00 D
                                                %  NO. OF BUFFERS IN UHW OF A3                                                   01145C00 D
                                                %  THIS ROUTINE ACCEPTS A VARIABLE                                               01146C00 D
                                                %  BUFFER LENGTH FOR CRT OUTPUT, 33 WORDS                                        01147C00 D
                                                %  FOR PRT OUTPUT                                                                01148C00 D
0306  4809 0640 0C90 00F0                       AMPCR = MIR               % STORE AMPCR INTO STACK                             01149C00 D
0307  4809 20C3 001C 00F0                       LIT = MAR2                                                                    0115CC0C D
0308  02C0 00C3 0C0C 00E0                       STACK = LIT                                                                   01151C00 D
0309  2F5C 00C0 CC9C C06C                       EOUTPUT - 1 = CPCR        % (STACK) = AMPCR                                   01152C00 D
030A  4809 0000 0000 20F0                       ASR                       % REFERENCE BR1                                     01153C00 D
030B  4809 0F43 0010 C0F0                       BMAR = BR2                % STORE BUFFER ADDRESS IN ER2                        01154C00 D
030C  4809 E000 9000 C0F0                       A3 R = A3                 % CLEAR LHW OF A3 FOR COUNT                         01155C00 D
030D  0000 0C03 0000 002C                       15 = SAR                                                                      01156C00 D
030E  4809 E0C1 1000 00F0                       A3 L = A3                                                                     01157C00 D
030F  05AC C0C0 0090 C060           COMRPT:     EMULIN - 1 = CPCR         % GET CONTENTS OF BUFFER ADDRESS WORD                01158C00 D
0310  4809 0C41 2020 00F0                       B L = A2                  % STORE CONTENTS IN UHW OF A2                       01159C00 D
0311  000C 0000 0030 0020                       COMP 16 = SAR                                                                 C116CC00 D
0312  4809 0F47 001C 00F0                       BMAR + 1 L = BR2          % GET ADJACENT WORD ADDR.                            01161C00 D
0313  0000 0003 0C9C 003C                       COMP 0 = SAR                                                                  01162C00 D
0314  058C 0000 0030 C060                       EMULIN - 1 = CPCR         % B = CONTENTS OF ADJACENT WORD                      01163CC0 D
0315  4809 CC5C 0030 C0F0                       A2 OR B = MIR             % COMBINED INTO ONE 32 BIT WORD                      01164C00 D
0316  4809 0F46 0B3C 00F0                       BMAR + 1 = B              % STORE NEXT WORD PAIR ADDR. IN B                    01165C00 D
0317  4809 0000 0000 20FC                       ASR                       % REFERENCE BR1                                     01166C00 D
0318  4809 0F43 801C 40F0                       BMAR R = MAR2,CSAR        % COMPRESSED WORD DESTINATION                        01167C00 D
0319  000C 00C0 0030 001C                       B = SAR                                                                       01168C00 D
031A  4809 0C41 0020 00F0                       B L = BR1                 % STORE WORD PAIR ADDR. IN B                         01169C00 D
031B  4809 C000 0090 24F0                       ASE                       % REFERENCE BR2                                     01170060 D
031C  4809 0F41 0010 00F0           EMULOUT:    BMAR L = BR2              % COMPRESSED WORD DESTINATION IN BR2                 01171C00 D
031D  05C0 C0C0 0C90 0060                       EMULOUT - 1 = CPCR                                                            C1172C0C D
031E  4809 0F40 2000 00F0                       BMAR = A2                 % SAVE DESTINATION ADDR. IN B                        01173C00 D
031F  4809 20C3 001C 00F0                       LIT + 1 = MAR2            % BUFFER LENGTH IN IOCW + 2                          01174C00 D
0320  032C 0003 0F00 00E0                       IOCW = LIT                                                                    01175000 D
0321  4809 0F46 0C1C 00F0                       BMAR + 1 = MAR2           % IOCW + 2 = MAR2                                    01176C00 D
0322  2F30 00C3 C090 0060                       EINPUT - 1 = CPCR         % B = BUFFER LENGTH                                  01177C00 D
0323  4809 0C00 000C 20F0                       ASR                       % REFERENCE BR1                                     01178C00 D
0324  4809 0F43 CC10 00F0                       BMAR = BR2                % TRANSFER NEXT WORD PAIR ADDR TO BR2                01179C00 D
0325  4809 CCC1 0020 00F0                       A2 + 1 L = BR1            % NEXT DESTINATION ADDR. IN BR1                      0118CC00 D
0326  0000 00C3 0C70 0030                       COMP 8 = SAR                                                                  01181C00 D
0327  4309 E0C0 1C00 00F0                       A3 + 1 = A3               % INCREMENT COUNTER ( < ? 16 - 1)                    01182C00 D
0328  4809 E001 2000 00F0                       A3 L = A2                 % ISOLATE COUNTER IN A2                              01183C00 D
0329  021C 0C03 0030 00AC                       COMP 16 = SAR; 33 = LIT                                                       01184C00 D
032A  4809 C0C0 A090 00F0                       A2 R = A2                                                                     01185C00 D
032B  3E59 CC52 0090 00FC                       IF LC2 THEN A2 EQL B; SET LC2; SKIP                                           01186F00 D
032C  4809 C152 0030 C0F0                       A2 EQL LIT                                                                    C1187C00 D
032D  5819 00C0 0000 00F0                       IF TRUE THEN SKIP                                                             C1188C0C D
032E  30E0 0000 0090 0040                       COMRPT - 1 = MPCR                                                             01189000 D
032F  4809 20C3 0C1C C0FC                       LIT = MAR2                % RESTORE AMPCR FROM STACK                           0119CC0C D
0330  02C0 00C3 0070 00E0                       STACK = LIT                                                                   01191C00 D
0331  2F3C C0C3 0C0C 0060                       EINPUT - 1 = CPCR                                                             01192C00 D
0332  4809 0C40 009C C0F0                       B = AMPCR                                                                     01193C00 D
0333  4809 0000 003C 00FC                       STEP                      % SEPARATE AMPCR ASSIGNMENTS AND JUMP                01194C00 D
0334  482D 0000 0000 C0FC                       JUMP                                                                          01195C00 D
                                                %                                                                             01196000 D
                                                %                                                                             01197C00 D
```

```
                        EXPAND:                 % THIS ROUTINE WILL TAKE A BUFFER       01198000 D
                                                % CONTAINING 32 BIT WORDS AND EXPAND EACH01199000 D
                                                % INTO 2 - 32 BIT WORDS, PUTTING THE INF01200000 D
                                                % IN THE LHW OF EACH 32 BIT WORD.        01201000 D
                                                % BR1 MUST CONTAIN THE BUFFER ADDRESS    01202000 D
                                                % AND A3 MUST CONTAIN # OF BUFFERS/COUNT 01203000 D
0335  4809 0000 0030 20F0          ASR           % REFERENCE BR1                         01204000 D
0336  4809 0F40 A030 00F0          DMAR R = A2                                          01205000 D
0337  0000 0000 003C 0010          B = SAR                                              01206000 D
0338  4809 E0C1 0D70 C0F0          A3 L = B      % ISOLATE THE COUNT(BUFF EXPAND FACTOR) 01207000 D
0339  0000 0000 0030 0020          COMP 16 = SAR                                        01208000 D
033A  4809 0C40 8B3C C0F0          D R = B                                              01209000 D
033B  4809 CC40 2000 00F0          A2 + B= A2    % CALCULATE FIRST OUTPUT ADDRESS       01210000 D
033C  4809 C0DE 2000 00F0          A2 - 1 = A2                                          01211000 D
033D  4809 0641 002C 00F0          AMPCR L = BR1 % RETURN IN BR1                        01212000 D
033E  0000 0000 003C 0030          COMP 8 = SAR                                         01213000 D
033F  4809 C001 0010 C0F0          A2 L = BR2    % BUFFER ADDR + BEF - 1 IN BR2         01214000 D
0340  05DC 0003 0000 0060  FXPRPT: EMULIN - 1 = CPCR % CONTENTS OF 1 WORD INTO C        01215000 D
0341  4809 0C41 2000 00F0          B L = A2      % WORD CONTENTS INTO LHW OF A2          01216000 D
0342  0000 0003 0000 0020          COMP 16 = SAR                                        01217000 D
0343  4809 C0C3 803C 24F0          A2 R = MIR; ASE                                      01218000 D
0344  4809 E001 2070 00F0          A3 L = A2     % ISOLATE BUFFER COUNT                 01219000 D
0345  4809 CC00 A030 00F0          A2 R = A2                                            01220000 D
0346  4809 CF41 0010 00FC          A2 + BMAR L = BR2 % BUFFER ADDR + COUNT INTO BR2     01221000 D
0347  0000 0000 0030 0030          COMP 8 = SAR                                         01222000 D
0348  05EC 0CC0 0000 C060          EMULOUT - 1 = CPCR % OUTPUT 1 EXPANDED HALFWORD      01223000 D
0349  4809 0000 0000 24F0          ASE           % REFERENCE MAR2                       01224000 D
034A  4809 0F40 2070 00FC          BMAR = A2                                            01225000 D
034B  4809 E0C1 0B00 00F0          A3 L = D      % ISOLATE THE COUNT                    01226000 D
034C  0000 0000 003C 0020          COMP 16 = SAR                                        01227000 D
034D  4809 0C40 8B0C 00FC          B R = D       % COUNT INTO B                         01228000 D
034E  4809 CC5F 0010 00F0          A2 - B L = BR2 % NEXT READ ADDRESS INTO BR2          01229000 D
034F  0000 00C0 0000 0030          COMP 8 = SAR                                         01230000 D
0350  05F0 0000 0030 C060          EMULIN - 1 = CPCR % NEXT WORD INTO B                 01231000 D
0351  4809 0C40 8C30 00FC          B R = MIR                                            01232000 D
0352  0000 00C0 0020 0020          16 = SAR                                             01233000 D
0353  4809 EC0F 2030 00F0          A3 - 1 L = A2 % REDUCE THE COUNT BY 1                01234000 D
0354  4809 C003 A030 00F0          A2 R = A2                                            01235000 D
0355  4809 CF41 0010 C0F0          A2 + BMAR L = BR2 % NEXT WRITE ADDRESS               01236000 D
0356  0000 0003 003C 0030          COMP 8 = SAR                                         01237000 D
0357  060C 0060 0000 0060          EMULOUT - 1 = CPCR % WRITE OUT NEXT EXPANDED WORD    01238000 D
0358  4809 E0DE 1C0C C0F0          A3 - 1 = A3   % REDUCE THE COUNT                     01239000 D
0359  4809 E0C1 2030 00F0          A3 L = A2                                            01240000 D
035A  0000 00C0 003C 0020          COMP 16 = SAR                                        01241000 D
035B  4809 C0C3 AB3C C0F0          A2 R = A2, B                                         01242000 D
035C  4809 C012 0000 00F0          A2 ECL 0      % CHECK IF ALL WORDS EXPANDED          01243000 D
035D  5019 0003 0000 00F0          IF FALSE THEN SKIP                                   01244000 D
035E  0610 00C3 0030 C040          EXPEND - 1 = MPCR                                    01245000 D
035F  4809 0F40 2000 00FC          BMAR = A2                                            01246000 D
0360  4809 CC59 0010 00F0          A2 - B- 1 L = BR2 % NEXT READ ADDRESS                01247000 D
0361  0000 0000 0030 0030          COMP 8 = SAR                                         01248000 D
0362  33F0 00C3 0000 C040          EXPRPT - 1 = MPCR                                    01249000 D
0363  4809 CCC0 003C 20F0  EXPEND: ASR           % REFERENCE MR1                        01250000 D
0364  4809 0F40 8040 C0F0          BMAR R = AMPCR % RESTORE RETURN ADDRESS              01251000 D
0365  0000 00C0 003C 0010          8 = SAR                                              01252000 D
0366  4809 00C0 003C C0F0          STEP                                                 01253000 D
0367  482D 0003 0000 00FC          JUMP                                                 01254000 D
                                                %                                      01255000 D
                                                %                                      01256000 D
                        EOUT:                   % THIS ROUTINE DUMPS THE BUFFERS        01257000 D
```

```
                                                    % INDICATED BY THE BUFFER ADDRESS TO     01258C00 D
                                                    % PRINTER, CRT, OR DISK (NOT IMPLEMENTED)01259C00 D
0368  3809 0000 0C3C 00F0          IF LC2            % (IOCW) = A2; MAR2 = IOCW                0125AC0C D
0369  4809 00C0 0000 24F0          ASE               % REFERENCE OR2                           01261000 D
036A  4809 0F46 0C1C 00F0          BMAR + 1 = MAR2   % IOCW + 1                               01262000 D
036B  2F30 00C0 0C00 0060          EINPUT - 1 = CPCR  % B = (IOCW+1)                          01263C00 D
036C  4809 0C41 0020 00F0          B L = BR1         % STORE (IOCW + 1) BUFFER ADDRESS IN BR1  01264000 D
036D  0000 00C0 0C3C C030          COMP 8 = SAR                                               01265C00 D
036E  4809 C0C0 A00C 00F0          A2 R = A2                                                  01266C00 D
036F  A000 0000 0020 0010          19 = SAR          % A2 HAS DEVICE CODE                     01267C00 D
0370  4809 0640 0C30 00F0          A2 + AMPCR = AMPCR                                         01268C0C D
0371  062C 0000 0030 C0C0          OUTDEV - 1 = AMPCR                                         01269000 D
0372  4809 0CC0 003C C0F0          STEP                                                       0126AC00 D
0373  4824 0000 0030 00FC          EXEC                                                       0126BC00 D
                                                    %                                        0126CC00 D
0374  063C 0000 C030 0040    OUTDEV: SPOO - 1 = MPCR                                        0126DC00 D
0375  0640 00C3 0C30 0040          PRTO - 1 = MPCR                                            0126EC00 D
0376  0650 00C0 003C 0040          NOTIMP - 1 = MPCR  % DISK IO NOT IMPLEMENTED               0126FC00 D
                                                    %                                        01270C00 D
                             SPOO:                  % WRITE SPO FUNCTION                     01271C00 D
0377  4849 0000 0000 00F0          SET LC2           % LC2 INDICATES IOCW + 1 WILL BE USED     01272C00 D
0378  305C 00C0 0C00 0060          COMPRES - 1 = CPCR % COMPRESS BUFFER TO 32 BIT I/O FORMAT  01273C0C D
0379  4809 2000 0C1C 00F0    BOSPO: LIT + 1 = MAR2    % BUFFER ADDRESS TO MIR                 01274C00 D
037A  032C 0000 C030 00E0          IOCW = LIT                                                 01275C00 D
037B  2F30 00C0 0030 0060          EINPUT - 1 = CPCR                                          01276C00 D
037C  4809 0C40 0C30 C0F0          B = MIR                                                    01277C00 D
037D  4809 20C1 0F00 00F0          LIT L = BB1                                                01278C00 D
037E  007C C000 0020 C0A0          7 = LIT; COMP 16 = SAR                                     01279C00 D
037F  4809 0C40 0030 24FC          B = MIR; ASE      % MIR CONTAINS 7/BUFFER ADDR.            0127AC00 D
0380  4809 0F46 0C1C 00F0          BMAR + 1 = MAR2   % BUFFER LENGTH INTO B (MAILBOX-1)       0127BC00 D
0381  2F3C C000 0C30 0060          EINPUT - 1 = CPCR                                          0127CC00 D
0382  4809 0C40 203C 00F0          B = A2            % TRANSFER BUFFER LENGTH INTO A2          0127DC00 D
0383  066C 0000 0030 C060          IOCIO - 1 = CPCR                                           0127EC00 D
0384  3780 0000 0000 C040          BOSPO - 1 = MPCR                                           0127FC00 D
0385  4809 0C00 0090 C0F0          A2 = MIR          % SAVE A2 IN MIR                         01280C00 D
0386  4809 E0C0 AC3C C0F0          A3 R = A2         % ISOLATE COUNTER                        01281C00 D
0387  000C 00C0 0000 0020          16 = SAR                                                   01282C00 D
0388  4809 CCDE 2C30 00F0          A2 - 1 = A2       % DECREMENT COUNTER                      01283C00 D
0389  4809 E0E1 903C 00F0          A3 C = A3         % DECREMENT COUNTER IN A3                01284C0C D
038A  4809 E0DE 100C 00F0          A3 - 1 = A3                                                01285000 D
038B  4809 E0E1 9C70 C0FC          A3 C = A3                                                  01286C00 D
038C  4809 CC12 0030 C0F0          A2 EQL 0                                                   01287C00 D
038D  6019 0F00 0020 00F0          IF FALSE THEN SKIP % TEST THE COUNTER FOR ZERO             01288C0C D
038E  067C C000 0030 004C          OPCODE - 1 = MPCR                                          01289C0C D
038F  4809 2000 001C 00FC    BPRTO: LIT + 1 = MAR2    % REGENERATE BUFFER ADDRESS IN B        0128AC00 D
0390  032C 0000 C09C 00E0          IOCW = LIT                                                 0128BC00 D
0391  2F3C 00C0 0C00 0060          EINPUT - 1 = CPCR                                          0128CC00 D
0392  4809 0C40 2D30 0CF0          B = A2, BMI       % B = BUFFER LENGTH, A2 = BUFFER ADDR     0128DC00 D
0393  4809 0C41 0B30 00F0          B L = B           % MULTIPLY B BY 2                         0128EC00 D
0394  900C 00C0 0070 0030          COMP 1 = SAR                                               0128FC00 D
0395  4809 0C41 0020 C0F0          A2 + B L = BR1    % NEW BUFFER ADDR IN BR1                  01290C00 D
0396  00C0 0C00 0030 0030          COMP 8 = SAR                                               01291C00 D
0397  4809 0C40 0090 00F0          A2 + B = MIR      % NEW BUFFER ADDRESS IN IOCW +1           01292C0C D
0398  2F5C 00C3 C030 006C          EOUTPUT - 1 = CPCR                                         01293C00 D
0399  376C 0000 0000 0040          SPOO - 1 = MPCR                                            01294C00 D
                                                    %                                        01295C00 D
                             PRTO:                  % WRITE PRT FUNCTION                     01296C00 D
039A  3050 0000 0000 0060          COMPRES - 1 = CPCR % COMPRESS THE BUFFER                   01297C00 D
039B  4809 20C3 001C C0F0    BPRTO: LIT + 1 = MAR2    % REGENERATE BUFFER ADDRESS             01298C0C D
039C  032C 00C3 C03C 00E0          IOCW = LIT                                                 01299C00 D
```

```
039D  2F30 00C0 0090 C060          EINPUT - 1 = CPCR                                  01318C00 D
039E  4809 0C40 0090 00F0          B = MIR                  % STORE BUFFER ADDRESS IN MIR 01319000 D
039F  4809 00C1 0F30 C0F0          1 L = BR1                                          01320000 D
03A0  0C00 0000 0000 0020          COMP 16 = SAR                                      01321C00 D
03A1  4809 0C40 0090 00F0          B = MIR                                            01322000 D
03A2  0680 0000 0000 0060          IOCIO - 1 = CPCR                                   01323000 D
03A3  39AC 0003 0080 0040          PPRIO - 1 = MPCR                                   01324000 D
03A4  4809 E0E0 A000 00FC          A3 R = A2                % ISOLATE COUNTER          01325C00 D
03A5  0000 C000 0000 002C          16 = SAR                                           01326000 D
03A6  4809 CC0E 2030 00F0          A2 - 1 = A2              % DECREMENT COUNTER        01327000 D
03A7  4809 EC01 9000 00F0          A3 C = A3                % DECREMENT COUNTER IN A3  01328C00 D
03A8  4809 E0DE 1000 C0F0          A3 - 1 = A3                                        01329000 D
03A9  40C9 E0C1 9000 C0FC          A3 C = A3                % RESTORE A3               01330C00 D
03AA  4809 C012 0C30 C0FC          A2 EQL 0                 % IS COUNTER = 0           01331000 D
03AB  6019 00C3 0C90 00F0          IF FALSE THEN SKIP                                 01332000 D
03AC  0690 0000 0000 0040          OPCODE - 1 = MPCR                                  01333000 D
03AD  4809 20C3 001C 00F0          LIT + 1 = MAR2           % REGENERATE BUFFER ADDRESS 01334C00 D
03AE  0320 0000 0090 00E0          IOCW = LIT                                         01335000 D
03AF  2F30 00C0 0000 C060          EINPUT - 1 = CPCR                                  01336000 D
03B0  4809 2C41 0020 00F0          LIT + B L = BR1          % WRITE NEW ADDRESS IN BR1 01337000 D
03B1  042C 00C0 0C00 00B0          66 = LIT, COMP B = SAR                             01338000 D
03B2  4809 2C40 0090 00F0          LIT + B = MIR            % WRITE NEW BUFFER ADDR IN IOCW +1 01339000 D
03B3  2F50 00C0 C00C C060          EOUTPUT - 1 = CPCR                                 01340C0C D
03B4  3990 00C3 0090 C040          PRIO - 1 = MPCR                                    01341C00 D
                                                    %                                 01342000 D
                                   IN:              % THIS ROUTINE INPUTS DATA FROM    01343C00 D
                                                    % PERIPHERAL DEVICES TO THE DESIGNATED 01344000 D
                                                    % BUFFERS. CRD READER, CRT, OR DISK (NI) 01345C00 D
03B5  3809 C0D0 0030 C0F0          IF LC2           % CLEAR LC2                        01346CC0 D
03B6  4809 C0C3 A030 06F0          A2 R = A2        % A2 = (IOCW)                      01347C0C D
03B7  A000 00C0 0090 0010          14 = SAR         % A2 CONTAINS DEVICE CODE          01348C00 D
03B8  4809 C640 0C30 C0F0          A2 + AMPCR = AMPCR                                 01349C00 D
03B9  06A0 0000 0C30 00C0          INDEV - 1 = AMPCR                                  01350C00 D
03BA  4809 0000 0090 00F0          STEP                                               01351000 D
03BB  4824 00C3 0C30 C0F0          EXEC                                               01352C00 D
                                                    %                                 01353000 D
03BC  06B0 0003 0030 0040          INDEV:  SPOI - 1 = MPCR                            01354C00 D
03BD  06CC 0000 0090 0040          CRDI - 1 = MPCR                                    01355C0C D
03BE  06D0 0003 0000 0040          NOTIMP - 1 = MPCR        % DISK NOT IMPLEMENTED    01356C0C D
                                                    %                                 01357C00 D
                                   SPOI:            % READ SPO FUNCTION                01358C00 D
03BF  4809 20C3 001C 00F0          LIT + 1 = MAR2   % RETRIEVE BUFFER ADDR.           01359000 D
03C0  032C 0900 0C00 C0EC          IOCW = LIT                                         01360C00 D
03C1  2F30 0003 0030 C060          EINPUT - 1 = CPCR                                  01361C00 D
03C2  4809 0C40 0090 00F0          B = MIR                  % STORE BUFFER ADDR IN MIR 01362C0C D
03C3  4809 0C41 0020 00F0          B L = BR1                % STORE ADDRESS IN BR1     01363C00 D
03C4  00C0 0003 0C0C 0030          COMP B = SAR                                       01364C00 D
03C5  4809 2001 CF00 00F0          LIT L = BBI                                        01365C00 D
03C6  0060 00C0 0030 C0AC          6 = LIT, COMP 16 = SAR                             01366C00 D
03C7  4809 0C40 0090 00F0          B = MIR                                            01367000 D
03C8  06E0 0003 0000 0060          IOCIO - 1 = CPCR                                   01368C00 D
03C9  3BF0 00C0 0C0C 004C          SPOI - 1 = MPCR                                    01369C0C D
03CA  4809 E0C0 9000 00FC          A3 R = A3                % ISOLATE COUNTER          01370000 D
03CB  0000 00C0 0C30 0020          16 = SAR                                           01371C00 D
03CC  4809 E0DE 1C00 C0F0          A3 - 1 = A3              % DECREMENT COUNTER        01372000 D
03CD  4809 E0C1 1000 00F0          A3 L = A3                % RESTORE A3, LHW CLEARED  01373C00 D
03CE  4809 E1EC 1000 00F0          A3 OR LIT = A3           % BUFFER EXPANSION FACTOR IN LHW OF A3 01374CC0 D
03CF  014C 00C0 0C0C 00E0          20 = LIT                                           01375C00 D
0300  334C 00C0 003C 0060          EXPAND - 1 = CPCR        % EXPAND BUFFER           01376C0C D
0301  4809 E0C3 A000 00FC          A3 R = A2                % ISOLATE COUNTER          01377000 D
```

```
03D3  4809 CC12 0000 00F0          A2 EQL 0                  % IS COUNTER ZERO?                  01379C00 D
03D4  6019 0003 0C00 C0F0          IF FALSE THEN SKIP                                            01380000 D
03D5  06F0 0000 0030 0040          OPCODE - 1 = MPCR                                             01381C00 D
03D6  4809 20C3 001C 00F0          LIT + 1 = MAR2            % REGENERATE BUFFER ADDR             01382C00 D
03D7  0320 00C3 0000 0GEC          IOCW = LIT                                                    01383C00 D
03D8  2F30 C0C0 0030 006C          EINPUT - 1 = CPCR                                             01384CC0 D
03D9  4809 2C40 003C 03FC          LIT + B = MIR                                                 01385C00 D
03DA  0280 G0C3 0000 00E0          40 = LIT                  % NEXT BUFFER ADDR IN MIR (WITH BEF)  01386C00 D
03DB  2F5C 0000 0C30 0060          EOUTPUT - 1 = CPCR                                            01387C00 D
03DC  38E0 00C3 0000 0040          SPOI - 1 = MPCR                                               01388C00 D
                                                             %                                   01389C00 D
                                   CRDI:                     % READ CARD READER FUNCTION          01390C00 D
03DD  4809 20C3 001C 00F0          LIT + 1 = MAR2            % RETRIEVE BUFFER ADDR.              01391C00 D
03DE  0320 0000 0030 C0EC          IOCW = LIT                                                    01392C0C D
03DF  2F30 0000 0C00 006C          EINPUT - 1 = CPCR                                             01393C00 D
03E0  4809 0C40 0090 00F0          B = MIR                   % STORE BUFFER ADDR IN MIR           01394CCC D
03E1  4809 0C41 0020 00F0          B L = BR1                 % STORE ADDRESS IN BR1               01395C00 D
03E2  0000 0000 0000 C030          COMP 0 = SAR                                                  01396C00 D
03E3  070C 00C0 0030 0060          IOCIO - 1 = CPCR                                              01397C00 D
03E4  3DC0 00C3 0030 C040          CRDI - 1 = MPCR                                               01398C0C D
03E5  4809 E0C0 900C C0F0          A3 R = A3                 % ISOLATE COUNTER                    01399C00 D
03E6  0000 0000 0000 0020          16 = SAR                                                      01400C00 D
03E7  4809 E0DE 100C 00F0          A3 - 1 = A3               % DECREMENT COUNTER                  01401C0C D
03E8  4809 E001 1000 00F0          A3 L = A3                 % RESTORE A3, LHW CLEARED            01402C00 D
03E9  4809 E15C 1C3C 00F0          A3 OR LIT = A3            % BUFFER EXPANSION FACTOR IN LHW OF A3  01403C00 D
03EA  0140 00C3 0C00 0DE0          20 = LIT                                                      01404C00 D
03EB  3340 00C0 0000 C060          EXPAND - 1 = CPCR         % EXPAND BUFFER                     01405C00 D
03EC  4809 E0C3 A030 C0F0          A3 R = A2                 % ISOLATE COUNTER                    01406C00 D
03ED  000C 0000 0C30 C020          16 = SAR                                                      01407C00 D
03EE  4809 CC12 0000 00F0          A2 EQL 0                  % IS COUNTER ZERO?                  01408000 D
03EF  6019 0003 0C00 00F0          IF FALSE THEN SKIP                                            01409C0C D
03F0  0710 C0C0 0C30 0D40          OPCODE - 1 = MPCR                                             01410000 D
03F1  4809 20C3 001C 00FC          LIT + 1 = MAR2            % REGENERATE BUFFER ADDR             01411C00 D
03F2  0320 0003 0000 00EC          IOCW = LIT                                                    01412000 D
03F3  2F30 0000 0C30 0060          EINPUT - 1 = CPCR                                             01413C00 D
03F4  4809 2C40 003C 00F0          LIT + B = MIR                                                 01414C00 D
03F5  028C 0000 0070 C0EC          40 = LIT                  % NEXT BUFFER ADDR IN MIR (WITH BEF)  01415C0C D
03F6  2F5C 00C0 0C00 0060          EOUTPUT - 1 = CPCR                                            01416C00 D
03F7  3DC0 0003 0070 0040          CRDI - 1 = MPCR                                               01417C00 D
                                                             %                                   01418C00 D
                                                             %                                   01419C00 D
                                   IOCIO:                    % THIS ROUTINE DOES EXTERNAL IO      01420C0C D
                                                             % INTERFACE BETWEEN THE UYK20 IOC    01421C00 D
                                                             % AND THE BURROUGHS IOP.  LC2 INDICATES  01422C00 D
                                                             % IF MAILBOX -1 IS TO BE PASSED      01423C00 D
03F8  1C88 0018 001C 00FC                SET GC2; WHEN GC2 THEN B111=MAR2                       01424C00 D
03F9  98C9 C0C9 001C 1CFC                MW2; B110=MAR2; IF SAI                                 01425C00 D
03FA  9C08 0C40 0090 00F0                WHEN SAI THEN D = MIR                                  01426C00 D
03FB  3A0B 0018 001C 1CF0                B111 = MAR2; IF LC2 THEN MW2 ELSE SKIP                 01427C00 D
03FC  9B18 C0C0 0000 00F0                WHEN SAI THEN SET INT; SKIP                            01428C0C D
03FD  4909 0000 0C30 00F0                SET INT                                                01429C00 D
03FE  D008 0003 0C70 C0F0                WHEN NOT INT THEN STEP                                 01430C00 D
03FF  DE08 00C0 0C30 CCF0                WHEN INT THEN DEX; MR2                                 01431CC0 D
0400  AEC8 0C42 0C90 00F0                WHEN RDC THEN NOT B; RESET GC2                         01432C00 D
0401  682F 0003 0000 C0F0                IF ABT THEN JUMP ELSE RETN                             01433C00 D
                                                             %                                   01434C0C D
                                                             %                                   01435C00 D
                                   % * * * * * * * * * * * * INDIRECT WORD ROUTINES * * * * * * * * * * * * *C1436000 D
                                                             %                                   01437CC0 D
```

```
                                                     * IW1 AT Y = :Y: + (R(M))              01439000 D
                                                     * LC2 INDICATES BYTE INSTRUCTION       01440000 D
                                                     * ROUTINES RETURN B = Y                01441000 D
0402  4809 2000 001C 00F0         LIT = MAR2                                                01442000 D
0403  02C0 0000 0000 00E0         STACK = LIT        * STORE A3 WITH RETURN ADDR. IN STACK  01443000 D
0404  4809 E000 0030 00F0         A3 = MIR                                                  01444000 D
0405  2F50 0000 0000 0060         EOUTPUT - 1 = CPCR * WRITE TO EMULATION RESERVE BUFFER    01445000 D
0406  0720 0000 0000 0060         IFETCH - 1 = CPCR                                         01446000 D
0407  4809 E001 1000 00F0         A3 L = A3          * CLEAR UHW OF A3                      01447000 D
0408  0000 0000 0000 0020         COMP 16 = SAR                                             01448000 D
0409  4809 E000 9000 00F0         A3 R = A3                                                 01449000 D
040A  4809 0C41 0B00 00F0         B L = B            * UHW OF A3 CONTAINS:Y:                01450000 D
040B  4809 EC5C 1000 00F0         A3 OR B = A3       * LHW OF A3 CONTAINS INSTRUCTION       01451000 D
040C  2380 0000 0000 0060         CONTENTSRM-1 = CPCR * B = (R(M))                          01452000 D
040D  4809 E000 AC00 00F0         A3 R = A2          * A2 CONTAINS :Y:                      01453000 D
040E  0000 0000 0000 0020         16 = SAR                                                  01454000 D
040F  4809 CC40 0B00 00F0         A2 + B = B         * CALCULATE ADDRESS OF Y               01455000 D
0410  0730 0000 0000 0040         INDIRWD - 1 = MPCR                                        01456000 D
                                                     *                                      01457000 D
                                                     *                                      01458000 D
                       IAWOI:                        * INDIRECT ADDRESSING WITHOUT INDEXING 01459000 D
                                                     * IW AT Y = :Y:                        01460000 D
                                                     * LC2 INDICATES BYTE INST.; ROUTINE    01461000 D
                                                     * RETURNS B = Y AND A2 = (R(M))        01462000 D
0411  4809 2000 001C 00F0         LIT = MAR2                                                01463000 D
0412  02C0 0000 0000 00E0         STACK = LIT        * WRITE A3 WITH RETURN IN UHW          01464000 D
0413  4809 E000 0030 00F0         A3 = MIR                                                  01465000 D
0414  2F50 0000 0000 0060         EOUTPUT - 1 = CPCR                                        01466000 D
0415  0740 0000 0000 0060         IFETCH - 1 = CPCR  * RETRIEVE :Y: FIELD                   01467000 D
0416  0750 0000 0000 0040         INDIRWD - 1 = MPCR                                        01468000 D
                                                     *                                      01469000 D
                                                     *                                      01470000 D
                       INDIRWD:                      * THIS ROUTINE READS IN IW1 AND TESTS  01471000 D
                                                     * IF CASCADED OR DIRECT ADDRESSING IS  01472000 D
                                                     * REQUIRED                             01473000 D
0417  4809 E001 1000 00F0         A3 L = A3          * CLEAR UHW OF A3                      01474000 D
0418  0000 0000 0000 0020         COMP 16 = SAR                                             01475000 D
0419  4809 E000 9000 00F0         A3 R = A3                                                 01476000 D
041A  4809 0C41 0010 00FC         B L = BR2                                                 01477000 D
041B  0000 0000 0000 003C         COMP 8 = SAR       * BR2 = IW1 ADDRESS                    01478000 D
041C  4809 0C41 0B00 00F0         B L = B                                                   01479000 D
041D  0000 0000 0000 0020         COMP 16 = SAR      * SAVE IW1 ADDR IN UHW OF A3           01480000 D
041E  4809 EC5C 1000 00FC         A3 OR B = A3                                              01481000 D
041F  0760 0000 0000 0060         EMULIN - 1 = CPCR  * B = (IW1)                            01482000 D
0420  4809 0C40 AC0C 00F0         B R = A2           * TEST IF BIT 14 OF IW1 IS SET         01483000 D
0421  A000 0000 0000 0010         14 = SAR                                                  01484000 D
0422  4809 C000 0000 00F0         A2                                                        01485000 D
0423  5800 0000 0000 00F0         IF LST THEN STEP ELSE SKIP                                01486000 D
0424  0770 0000 0000 004C         CASCADED - 1 = MPCR * IF BIT SET, CASCADED IW             01487000 D
0425  0780 0000 0000 0040         DIRECT - 1 = MPCR  * IF BIT NOT SET, FORM Y               01488000 D
                                                     *                                      01489000 D
                                                     *                                      01490000 D
                       CASCADED:                     * THIS ROUTINE DECIDES WHICH ADDRESSING 01491000 D
                                                     * FORMULA TO USE TO COMPUTE THE NEW    01492000 D
                                                     * LOCATION OF THE INDIRECT WORD        01493000 D
0426  4809 18C1 AC00 00F0         B101 C = A2        * A2 IS BIT MASK FOR J FIELD OF IW     01494000 D
0427  3000 0000 0000 0020         19 = SAR                                                  01495000 D
0428  4809 CC56 A000 00FC         A2 AND B R = A2                                           01496000 D
0429  3000 0000 0000 0010         12 = SAR                                                  01497000 D
```

```
042A  4809 C640 0C9C 00F0          A2 + AMPCR = AMPCR     % COMPUTE FORMULA DESIRED          01498C00 D
042B  079C 0003 0C0C C0C0          IWADDR - 1 = AMPCR                                        01499C0C D
042C  4809 0000 0F30 00F0          STEP                                                      01500C00 D
042D  4824 0000 0090 00F0          EXEC                                                      01501C00 D
                                                                                             01502000 D
                                                                                             01503000 D
042E  07A0 0000 0000 0040  IWADDR: ADT0 - 1 = MPCR                                           01504000 D
042F  0780 0000 0C30 0040          ADT1 - 1 = MPCR                                           01505C00 D
0430  07CC 0CC0 0C00 0040          ADT2 - 1 = MPCR                                           01506C00 D
0431  070C C000 0000 C040          ADT3 - 1 = MPCR                                           01507000 D
                                                                                             01508C00 D
                                                                                             01509C00 D
                           ADT0:                                 % NEW IW AT Y = IW2        01510C0C D
0432  4809 E000 A00C 00F0          A3 R = A2                     % SHIFT ADDR OF IW1 INTO A2 01511C00 D
0433  000C 0000 0070 0020          16 = SAR                                                  01512C00 D
0434  4809 C0D3 0C10 00F0          A2 OR 1 L = BR2               % NEXT ADDRESS OF IW        01513C00 D
0435  000C 0000 0030 003C          COMP 8 = SAR                                              01514C00 D
0436  07EC 0000 000C C060          EMULIN - 1 = CPCR             % R CONTAINS IW2; Y = IW2   01515C00 D
0437  416C 00C0 0000 0040          INDIRWD - 1 = MPCR            % Y = NEXT ADDR OF IW1      01516C00 D
                                                                                             01517C00 D
                                                                                             01518000 D
                           ADT1:                                 % NEW IW AT Y = IW2 + (R(X)) 01519C00 D
0438  4809 2C56 0B3C 00F0          B AND LIT = B                                             01520C00 D
0439  00F0 000C 0C0C 00EC          15 = LIT                      % CREATE REGISTER R(X)      01521CC0 D
043A  07F0 00C0 0030 0060          REGSTACK - 1 = CPCR                                       01522C0C D
043B  2F3C 0003 0030 0060          EINPUT - 1 = CPCR             % B = (R(X))               01523C00 D
043C  0800 0003 003C C040          ADYF - 1 = MPCR                                           01524C0C D
                                                                                             01525C00 D
                                                                                             01526C00 D
                           ADT2:                                 % NEW INDIRECTWORD AT Y = IW2 + (R(M)) 01527C00 D
043D  2380 0003 0030 0C60          CONTENTSRM - 1 = CPCR  % A3 HAS INST. IN LOWER 16 BITS     01528C00 D
                                                         % B = (R(M))                        01529C00 D
043E  0810 0C00 0000 0040          ADYF - 1 = MPCR                                           01530C0C D
                                                                                             01531C00 D
                                                                                             01532C0C D
                           ADT3:                                 % NEW IW AT Y = IW2 + (R(M+1)) 01533CC0 D
043F  4809 E156 0B3C 00F0          A3 AND LIT = B                                            01534C00 D
0440  00FC 0000 0C3C 0CE0          15 = LIT                                                  01535C00 D
0441  4809 0C46 0B3C C0F0          B + 1 = B                     % B = RM+1                  01536C00 D
0442  0820 0000 C000 C060          REGSTACK - 1 = CPCR                                       01537C00 D
0443  2F3C 0C03 003C C060          EINPUT - 1 = CPCR             % B = (R(M+1))              01538C00 D
0444  083C C0C0 0C30 C040          ADYF - 1 = MPCR                                           01539CC0 D
                                                                                             01540C00 D
                                                                                             01541C00 D
                           ADYF:                                 % THIS ROUTINE COMPUTES Y GIVEN 01542CC0 D
                                                                 % B = (R(X)),(R(M)), OR (R(M+1)) 01543C00 D
0445  4809 E0C0 A00C 0CF0          A3 R = A2                     % SHIFT ADDR OF IW INTO A2  01544C0C D
0446  000C 0000 0C30 0020          16 = SAR                                                  01545C00 D
0447  4809 CCDD 0C10 00F0          A2 OR 1 L = BR2               % NEXT ADDR OF IW           01546CC0 D
0448  000C C0C0 00FC 0030          COMP 8 = SAR                                              01547C00 D
0449  4809 0C40 200C 00FC          B = A2                        % SAVE (R(X)),(R(M)), OR (R(M+1)) IN A2 01548C00 D
044A  084C C003 003C 0060          EMULIN - 1 = CPCR             % B = (IW2)                 01549CC0 D
044B  4809 CC40 0B3C 00F0          A2 + B = B                    % Y = (IW2) + ((R(X)),(R(M)), OR (R(M+1)) 01550C00 D
044C  416C 0000 00C0 0040          INDIRWD - 1 = MPCR            % Y = NEXT ADDR OF IW1       01551C0C D
                                                                                             01552000 D
                                                                                             01553C0C D
                           DIRECT:                               % THIS ROUTINE DECIDES WHAT ADDRESSING 01554C00 D
                                                                 % FORMULA TO COMPUTE Y - EFF. OPERAND 01555C00 D
                                                                 % ADDRESS                    01556C0C D
044D  4809 18C1 AC30 00F0          B101 C = A2                                               01557C00 D
```

```
044E  5000 0000 0000 0020          19 = SAR                  % MASK :J: FIELD OF IW1          01558C00 D
044F  4809 CC56 A090 00F0          A2 AND B R = A2                                            01559000 D
0450  5000 0000 0000 6010          12 = SAR                  % ISOLATE BIT VALUE IN A2        01560C00 D
0451  4809 C640 0C90 C0F0          A2 + AMPCR = AMPCR        % COMPUTE JUMP TABLE ADDRESS      01561000 D
0452  085C 00C0 0090 C0C0          DIRADDR - 1 = AMPCR       % FOR FORMULA DESIRED             01562C00 D
0453  4809 0000 0000 00F0          STEP                                                       01563C00 D
0454  4824 0000 0C90 00F0          EXEC                                                       01564000 D
                                                             %                                01565C0C D
                                                             %                                01566C00 D
0455  0860 0003 0070 0040    DIRADDR: DI0 - 1 = MPCR                                         01567C00 D
0456  0870 0000 0090 0040             DI1 - 1 = MPCR                                         01568000 D
0457  0880 0000 0C00 0040             DI2 - 1 = MPCR                                         01569000 D
0458  0890 0003 0000 0040             DI3 - 1 = MPCR                                         0157 0C0C D
                                                             %                                01571C00 D
                                                             %                                01572C00 D
                                       DI0:                  % Y = IW2                        01573000 D
0459  4809 E0C3 AC90 00FC             A3 R = A2               % SHIFT ADDR OF IW1 INTO A2       01574C00 D
045A  0000 0003 0070 0020             16 = SAR                                               01575CC0 D
045B  4809 CC03 0C10 C0F0             A2 OR 1 L = BR2                                        01576C00 D
045C  0000 0000 0030 0030             COMP B = SAR                                           01577C00 D
045D  38A0 0003 0000 0060             EMULIN - 1 = CPCR       % B = (IW2)                    01578C90 D
045E  3809 0000 0C30 00F0             IF LC2                  % CLEAR BYTE BIT(IF SET) LC2 = MS BYTE  01579C0C D
045F  0880 0000 0C30 0040             DIRETURN - 1 = MPCR                                    01580C00 D
                                                             %                                01581C0C D
                                                             %                                01582C00 D
                                       DI1:                  % Y = IW2 + (R(X))              01583000 D
0460  4809 2C56 0B30 00FC             B AND LIT = B                                          01584C00 D
0461  00F0 0000 0000 00E0             15 = LIT                % ISOLATE RX FROM IW1           01585C00 D
0462  08C0 C003 CC0C C060             REGSTACK - 1 = CPCR                                    01586C00 D
0463  2F30 0003 0000 0060             EINPUT - 1 = CPCR       % B = (R(X))                   01587C0C D
0464  08D0 0CC0 0C30 0040             DIGN - 1 = MPCR                                        01588C00 D
                                                             %                                01589000 D
                                                             %                                01590C00 D
                                       DI2:                  % Y = IW2 + (R(M))              01591C00 D
0465  4809 E156 0B30 00F0             A3 AND LIT = B                                         01592C00 D
0466  00F0 0000 CC30 00E0             15 = LIT                % MASK OFF :M: FIELD :M: = RM    01593C00 D
0467  08E0 0003 0C00 C060             REGSTACK - 1 = CPCR                                    01594C00 D
0468  2F30 CC03 CC00 006C             EINPUT - 1 = CPCR       % B = (R(M))                   01595C00 D
0469  08F0 00C0 003C C040             DIGN - 1 = MPCR                                        01596C00 D
                                                             %                                01597C00 D
                                                             %                                01598000 D
                                       DI3:                  % Y = IW2 + (R(M+1))            01599C00 D
046A  4809 E156 0B3C 00F0             A3 AND LIT = B                                         016C0000 C
046B  00F0 00C3 0030 C0E0             15 = LIT                % MASK OFF RM INTO B            01601C00 D
046C  4809 0C46 0B90 00FC             B + 1 = B               % B = RM + 1                   01602CCC D
046D  090C 0C03 003C 0060             REGSTACK - 1 = CPCR                                    016C3C00 D
046E  2F3C 0003 0030 0060             EINPUT - 1 = CPCR       % P = (R(M+1))                 01604C0C D
046F  0910 C0C3 003C 0040             DIGN - 1 = MPCR                                        01605C00 D
                                                             %                                016C6C90 D
                                                             %                                01607C00 D
                                       DIGN:                 % THIS ROUTINE COMPUTES Y, TEST FOR BYTE 01608000 D
                                                             % INST. AND RETURNS TO CALLER OF 01609C0C D
                                                             % RXMFIELD                       01610CC0 D
0470  4809 E0C3 AC9C C0F0             A3 R = A2               % IW1 ADDR INTO BR2 INC. BY 1   01611C0C D
0471  0000 00C3 0000 C020             16 = SAR                                               01612000 D
0472  4809 C0D3 0010 CCFC             A2 OR 1 L = BR2                                        01613C00 D
0473  0000 0003 003C 003C             COMP B = SAR                                           01614C0C D
0474  4809 0C40 2020 00F0             B = A2                  % SAVE (R(X)),(R(M)), OR(R(M+1)) IN A2  01615C00 D
0475  0920 0003 0C3C 0060             EMULIN - 1 = CPCR       % B = (IW2)                    01616C00 D
0476  4809 C0C3 0C9C 00F0             A2 = MIR                                               01617CC0 D
```

```
0477  4809 0C40 2030 00F0          B = A2,BMI          % A2 = (W2); B =(R(R)),(R(X)),(R(P+1))    01618000 D
0478  3A06 0000 0030 00FC          IF LC2 THEN SET LC2;STEP ELSE SKIP                           01619000 D
0479  1E1C 00C0 0000 006C          BYTETEST - 1 = CPCR   % DO BYTE FUNCTION IF LC2 SET          01620C00 D
047A  4809 CC40 C830 00F0          A2 + B = B            % B = Y                                01621000 D
047B  093C 0000 C000 0040          DTRETURN - 1 = MPCR                                        01622000 D
                                                         %                                    01623000 D
                                                         %                                    01624000 D
                                   DTRETURN:             % THIS ROUTINE RETURNS TO CALLER      01625000 D
                                                         % OF BXMFIELD; ENTERS WITH B = Y      01626000 D
047C  4809 0C40 2000 00F0          B = A2                % A2 = Y                              01627000 D
047D  4809 20C3 0C1C C0F0          LIT = MAR2                                                 01628000 D
047E  02CC 0000 0030 00A0          STACK = LIT; 16 = SAR                                      01629000 D
047F  2F3C C0C0 0030 C060          EINPUT - 1 = CPCR     % READ IN CONTENTS OF  STACK         01630000 D
0480  4809 0C40 8040 00F0          B R = AMPCR           % RESTORE RETURN ADDRESS IN AMPCR     01631000 D
0481  4809 CC00 0B30 00F0          A2 = B                % P = Y                               01632000 D
0482  482D 0000 0C30 00F0          JUMP                  % JUMP TO BXMFIELD CALLER             01633000 D
                                                         %                                    01634000 D
                                   * * * * * * * * * END INDIRECT WORD ROUTINES * * * * * * * * * * * * * * * 01635000 D
                                                         %                                    01636000 D
                                   LDBLE:                % LOAD DOUBLE;                        01637000 D
                                                         % THIS ROUTINE TAKES AS INPUT AN ADDR. 01638000 D
                                                         % IN B AND STORES CONTENTS OF B AND B+1 01639000 D
                                                         % INTO RA,RA+1                        01640000 D
0483  4809 E001 100C 00F0          A3 L = A3                                                  01641000 D
0484  000C 00C3 0030 0020          COMP 16 = SAR                                              01642000 D
0485  4809 E65C 1000 C0F0          A3 OR AMPCR = A3      % SAVE RETURN IN UNW OF A3           01643000 D
0486  4809 E001 903C C0F0          A3 C = A3                                                  01644000 D
0487  4809 0C40 2000 C0F0          B = A2                % SAVE Y OR Y* IN A2                  01645000 D
0488  4809 0C41 CC10 C0F0          B L = BR2                                                  01646000 D
0489  000C 00C3 CC3C C030          COMP B = SAR                                               01647000 D
048A  094C C0C0 0C00 0060          EMULIN - 1 = CPCR     % (Y) OR (Y*) INTO B                 01648000 D
048B  4809 C0D3 001C C0F0          A2 OR 1 L = BR2       % DR2 HAS Y* +1 OR Y+1               01649000 D
048C  0000 C0C0 0C30 0030          COMP B = SAR                                               01650000 D
048D  4809 0C41 2000 C0F0          B L = A2              % A2 = (Y*) UNW OR (Y) UNW           01651000 D
048E  0000 C0C0 00CC 0620          COMP 16 = SAR                                              01652000 D
048F  095C 0000 0C0C 0060          EMULIN - 1 = CPCR     % B = (Y*+1) OR (Y+1)                01653000 D
0490  4809 CC5C 2030 00F0          A2 OR B = A2          % A2 = (Y*)/(Y*+1) OR (Y)/(Y+1)       01654000 D
0491  4809 E0C0 8D30 00FC          A3 R = B              % MASK OFF :A: FIELD                  01655000 D
0492  80F0 0C03 0C00 0080          4 = SAR; 15 = LIT                                          01656000 D
0493  4BC9 2C55 0B0D 00F0          LIT AND B = B         % :A: FIELD INTO B                   01657000 D
0494  4809 C000 B03C C0F0          A2 R = MIR            % (Y) OR (Y*) INTO MIR               01658000 D
0495  000C 00C0 0C0C 0020          16 = SAR                                                   01659000 D
0496  0960 00C0 0C30 0060          REGSTACK - 1 = CPCR   % R(A) INTO MAP2                     01660000 D
0497  2F50 00C0 0000 C060          EOUTPUT - 1 = CPCR    % (Y) OR (Y*) INTO R(A)              01661000 D
0498  4809 C0C0 0B0C C0F0          A2 = B                % SET DOUBLEWORD ARITH COMPARE        01662000 D
0499  49C9 60C0 0000 00F0          SET LC1                                                    01663000 D
049A  20D0 0300 0C0C 006C          SETCCA - 1 = CPCR                                          01664000 D
049B  4809 C0C1 2030 00F0          A2 L = A2                                                  01665000 D
049C  001C 00D3 0000 00A0          COMP 16 = SAR;1 = LIT                                      01666000 D
049D  4809 2F5C 0C1C C0FC          LIT OR BMAR = MAR2    % R(A+1) INTO MAR2                   01667000 D
049E  4BC9 C0D3 B030 00F0          A2 R = MIR            % (Y*+1) INTO MIR                    01668000 D
049F  2F50 00D3 C0CC 0060          EOUTPUT - 1 = CPCR                                         01669000 D
04A0  4809 E0C0 8C40 00F0          A3 R = AMPCR          % RETURN ADDRESS IN AMPCR            01670000 D
04A1  4809 0C03 0030 0CF0          STEP                                                       01671000 D
04A2  482D 00C0 0030 00F0          JUMP                                                       01672000 D
                                                         %                                    01673000 D
                                                         %                                    01674000 D
                                   MOVE:                 % MOVES THE NUMBER OF CELLS SPECIFIED 01675000 D
                                                         % BY UNW OF A3 FROM ADDR SPECIFIED BY ER1 01676000 D
                                                         % INTO ADDR  LOCATED IN LHW OF A3     01677000 D
```

```
                                                    % PAGE ADDRESSING IS ALLOWED                    01678000 D
04A3  4809 C640 2000 00F0           AMPCR = A2      % SAVE RETURN IN A2                             01679000 D
04A4  4809 0000 0000 20F0   MRPT:   ASR                                                            01680000 D
04A5  4809 0F40 0010 00F0           BMAR = BR2      % BR1 = BR2                                     01681000 D
04A6  0970 0F00 0000 C060           EMULIN - 1 = CPCR % B = (BR1)                                   01682000 D
04A7  4809 0C40 0C90 00F0           B = MIR         % TRANSFER B TO MIR                             01683000 D
04A8  4809 C000 0E90 00F0           A2 = B          % STORE AMPCR IN B                              01684000 D
04A9  4809 E001 0010 00F0           A3 L = BR2      % LHW OF A3 INTO BR2                            01685000 D
04AA  0C00 00C0 00DC 0030           COMP B = SAR                                                   01686000 D
04AB  4809 E000 9090 00F0           A3 R = A3       % ZERO OUT LHW OF A3                            01687000 D
04AC  000F 0000 0090 0020           16 = SAR                                                       01688000 D
04AD  4809 E0C1 100C 00F0           A3 L = A3                                                      01689000 D
04AE  4809 EC5C 1000 00F0           A3 OR B = A3    % STORE AMPCR IN LHW OF A3                     01690000 D
04AF  0980 0000 0090 0060           EMULOUT - 1 = CPCR                                             01691000 D
04B0  4809 E0C1 2000 00FC           A3 L = A2       % STORE AMPCR IN UHW OF A2                     01692000 D
04B1  0000 00F0 009C 0020           COMP 16 = SAR                                                  01693000 D
04B2  4809 C0C3 AF90 00F0           A2 R = A2       % RETURN AMPCR TO LHW OF A2                    01694000 D
04B3  4809 E0E0 9090 00F0           A3 R = A3       % ZERO LHW OF A3                               01695000 D
04B4  4809 E0C1 100C 00FC           A3 L = A3                                                      01696000 D
04B5  4809 0F46 0B9C 00F0           BMAR + 1 = B    % INCREMENT ADDRESS DESTINATION                01697000 D
04B6  4809 EC5C 1090 20F0           A3 OR B = A3; ASR % STORE DESTINATION ADDR IN LHW OF A3         01698000 D
04B7  4809 0F40 8D9C 40F0           BMAR R = B,CSAR                                                01699000 D
04B8  00C0 00C0 009C 0010           B = SAR                                                        01700000 D
04B9  4809 0C47 002C 00F0           B + 1 L = BR1   % INCREMENT ORIGIN                             01701000 D
04BA  4809 E001 9090 00FC           A3 C = A3       % SHIFT COUNT TO LHW OF A3                     01702000 D
04BB  0000 0000 0090 0020           16 = SAR                                                       01703000 D
04BC  4809 E3DE 1B90 00F0           A3 - 1 = A3,B   % DECREMENT COUNTER                            01704000 D
04BD  4809 0C41 0B0C 00F0           B L = B         % ISOLATE COUNTER IN LHW OF B                  01705000 D
04BE  4809 0C40 8B00 00F0           B R = B                                                        01706000 D
04BF  4809 E001 9090 00F0           A3 C = A3       % RESTORE A3                                   01707000 D
04C0  4809 0C52 0090 00F0           B EQL 0                                                        01708000 D
04C1  0000 00C0 00DC 0030           COMP B = SAR                                                   01709000 D
04C2  600B 0000 0000 00F0           IF FALSE THEN STEP ELSE SKIP                                   01710000 D
04C3  4A3C 0000 0090 C040           MRPT - 1 = MPCR                                                01711000 D
04C4  4809 C000 0C4C 00F0           A2 = AMPCR                                                     01712000 D
04C5  48C9 0000 0C90 00F0           STEP                                                           01713000 D
04C6  482D 0000 009C 00F0           JUMP                                                           01714000 D
                                                    %                                              01715000 D
                                                    %                                              01716000 D
                                    MULT:           % THIS ROUTINE WILL MULTIPLY 2 VALUES.          01717000 D
                                                    % THE OPERANDS MUST BE PASSED BY A2 AND B      01718000 D
                                                    % LC1 MUST BE SET IF 32 BIT NUMBERS ARE        01719000 D
                                                    % PASSED.  PRODUCT IS RETURNED IN A3.          01720000 D
04C7  4809 0F00 1000 00F0           C = A3          % HOLDS PARTIAL PRODUCT                         01721000 D
04C8  0000 0003 0090 0020           16 = SAR                                                       01722000 D
04C9  28D9 00F3 0090 00F0           IF LC1 THEN SET LC1; SKIP                                      01723000 D
04CA  099C 00F3 0C90 0040           SREG - 1 = MPCR     % DOUBLE PACK SINGLE REG VALUES            01724000 D
04CB  4809 0C52 0090 00F0   CMUL:   C EQL B                                                        01725000 D
04CC  64DD CC4C 0000 0CFC           IF FALSE THEN A2 XOR B; STEP ELSE JUMP  % CHECK FOR NEG PROD   01726000 D
04CD  4A49 C0F0 003C 00FC           A2; IF MST THEN SET LC2 % FLAG FOR NEG PRODUCT                 01727000 D
04CE  4CCB CFF2 2000 C0F0           IF MST THEN NOT A2 = A2; STEP ELSE SKIP                        01728000 D
04CF  48C9 C0C0 2000 F0F0           A2 + 1 = A2         % TWO'S COMPLEMENT                         01729000 D
04D0  4809 0C40 0C90 00F0           B                   % BRING B TO THE ADDER                     01730000 D
04D1  9C09 0C5E 0B90 00F0           IF MST THEN 0 - B = B  % COMPLEMENT B                          01731000 D
04D2  2819 C0C3 0090 00F0           IF LC1 THEN SKIP                                               01732000 D
04D3  09AC 00E0 0000 0040   \       RREG - 1 = MPCR     % RESTORE SINGLE REG TO LS WORDS           01733000 D
04D4  48C9 C0C3 0090 F0F0   MULT1:  A2                  % BRING A2 TO THE ADDER                    01734000 D
04D5  5C09 EC40 1000 00F0           IF LST THEN A3 + B = A3                                        01735000 D
04D6  4809 CFF0 A09C 40FC           A2 R = A2, CSAR     % SHIFT OFF LS BIT OF A2                   01736000 D
04D7  1000 0000 0070 0000           1 = SAR                                                        01737000 D
```

```
04D8  4809 0C41 0B30 00F0          B L = B                    % MULT MULTIPLIER BY 2          01738C00 D
04D9  4809 C00D 0030 00F0          A2                         % CHECK TO SEE IF ENTIRE MULTIPLIER 01739F00 D
04DA  4809 C012 0C30 00F0          A2 EOL 0                   % HAS BEEN READ                 0174CC00 D
04DB  6819 0000 0000 00F0          IF TRUE THEN SKIP          % IF MULTIPLICAND = 0, THEN DONE 01741C00 D
04DC  4D3C 0000 0030 0040          MULT1 - 1 = MPCR                                           01742C00 D
04DD  3C00 E002 1C00 00F0          IF LC2 THEN NOT A3 = A3;STEP ELSE JUMP                     01743C00 D
04DE  4809 E0C3 1000 00F0          A3 + 1 = A3                                                01744F00 D
04DF  4820 0003 0030 00F0          JUMP                                                       01745C00 D
04E0  4809 0C41 0B30 C0F0  SREG:   B L = B                    % MOVE B TO UHW                 0174600C D
04E1  4809 C0C1 2030 00F0          A2 L = A2                  % MOVE A2 TO UPW, 16 = SAR      01747C00 D
04E2  4CA0 0000 0000 0040          CMUL - 1 = MPCR            % CONTINUE MULT OP              01748000 D
04E3  4809 C0C0 A00C 00F0  RREG:   A2 R = A2                  % RESTORE REG TO LHW            01749C00 D
04E4  4809 0C40 8B0C C0F0          B R = B                    % RESTORE REG TO LHW            01750C00 D
04E5  403C 0000 0000 C040          MULT1 - 1 = MPCR                                           01751C00 D
                                                              %                              01752C00 D
                                                              %                              01753600 D
                                   NOTIMP:                    % THIS ROUTINE SETS UP AN ERROR MESSAGE 01754C00 D
                                                              % "NOT IMPLEMENTED" INSERTED VIA LOADER 01755C00 D
                                                              % JCL                           01756C00 D
04E6  0980 0003 0000 0060          SAVEREG - 1 = CPCR                                         01757000 D
04E7  4809 0640 0B3C C0F0          AMPCR = B                                                  01758C00 D
04E8  0CE0 0000 0030 00C0          ERRORLIST = AMPCR                                          01759C00 D
04E9  4809 2C41 0020 00F0          LIT + 0 L = BR1            % CREATE ERRORLIST ADDR. IN BR1 0176CC00 D
04EA  0080 00C0 0C30 0080          8 = LIT; COMP 8 = SAR                                      01761C00 D
04EB  09C0 0000 0C30 0040          PRTERR - 1 = MPCR                                          01762C00 D
                                                              %                              01763000 D
                                                              %                              01764C00 D
                                   OVERFLOW:                  % THIS ROUTINE  WILL PRINT OUT AN ERROR 01765C00 D
                                                              % DIAGNOSTIC INSERTED VIA LOADER JCL AND 0176600C D
                                                              % ABORTS MACHINE EXECUTION.     01767C00 D
04EC  090C 0C00 0030 0060          SAVEREG - 1 = CPCR                                         01768C0C D
04ED  4809 0640 0B3C 00F0          AMPCR = B                  % TRANSFER ERRORLIST ADDR TO B  01769C00 D
04EE  0CE0 0000 C030 00C0          ERRORLIST = AMPCR                                          0177CC00 D
04EF  4809 2C41 0C20 C0F0          LIT + B L = BR1            % CREATE ERRORLIST ADDR IN BR1  01771C0C D
04F0  010C 0000 003C C080          16 = LIT; COMP 8 = SAR                                     01772C00 D
04F1  09E0 0C00 0030 0040          PRTERR - 1 = MPCR          % PRINT THE ERROR MESSAGE AND STOP 01773C00 D
                                                              %                              01774C00 D
                                   * * * * * * * * * * * * OVERFLOW BIT SETTING * * * * * * * * * * * * 01775C0C D
                                                              %                              01776C0C D
                                   CHECKOV:                   % THIS ROUTINE WILL SET THE OVERFLOW BIT 01777C00 D
                                                              % IF REQUIRED, OTHERWISE CLEAR IT 01778C00 D
                                                              % A2 HOLDS "X", B HOLDS "Y", AND MIR 01779C0C D
                                                              % HOLDS THE ALG SUM (IN MS WORDS) 0178CC00 D
                                                              % IF LC2 IS SET, IT IMPLIES A NEGATIVE 01781C0C D
04F2  380B 00C3 0030 00FC          IF LC2 THEN STEP ELSE SKIP  % TEST FOR A SUBTRACTION OPCODE 01782C00 D
04F3  4809 0C5E 0B3C 00F0          C - B = E                                                  01783000 D
04F4  4809 C00D 003C C0F0          A2                         % A2 TO THE ADDER               01784C0C D
04F5  4BC9 00C0 0000 00FC          IF MST THEN SET LC1        % IMPLIES X IS NEGATIVE         01785C00 D
04F6  4809 CC4C 0C30 00F0          A2 XOR B                   % CHECK FOR "LIKE" SIGNS        01786000 D
04F7  480B 0C00 0C30 C0F0          IF MST THEN STEP ELSE SKIP                                 01787C00 D
04F8  09FC 00C3 000C 0040          CLEAROV - 1 = MPCR         % CLEAR THE OV BIT, UNLIKE SIGNS 01788C00 D
04F9  4809 00C3 0C3C 00F0          BMI                                                        01789000 D
04FA  4809 0C40 0030 00F0          B                                                          01790C0C D
04FB  480B 00C3 0F00 C0F0          IF MST THEN STEP ELSE SKIP  % CHECK FOR NEG SUM            01791C0C D
04FC  0A0C C0C3 0030 004C          SNEG - 1 = MPCR            % NEG SUM, LIKE SIGNS           01792C00 D
04FD  280B C0D0 0030 00F0          IF LC1 THEN STEP ELSE SKIP                                 01793C00 D
04FE  0A1C C0D3 0000 0040          SETOVBIT - 1 = MPCR        % POS SUM, LIKE SIGNS, NEG X    01794C00 D
04FF  0A2C 00C0 003C 0040          CLEAROV - 1 = MPCR         % POS SUM, LIKE SIGNS, POS X    01795C0C D
0500  280B 00C3 0C30 00FC  SNEG:   IF LC1 THEN STEP ELSE SKIP                                 01796C00 D
0501  0A3C 00C3 0C30 0040          CLEAROV - 1 = MPCR         % NEG SUM, NEG X, LIKE SIGNS    01797C0C D
```

```
0502  0A40 0000 0000 0040          SETOVBIT - 1 = MPCR   % NEG SUM, LIKE SIGNS, POS X          01798000 D
                                                         %                                    01799000 D
                                                         %                                    01800000 D
0503  2800 0000 0000 00F0   OVOIT:  IF LC1 THEN STEP ELSE SKIP  % THIS ROUTINE WILL SET THE    01801000 D
0504  0A50 0000 0020 0040          SETOVBIT - 1 = MPCR   % OVERFLOW BIT IN SR#1 IF LC1 IS SET  01802000 D
0505  0A60 0000 0000 0040          CLEAROV - 1 = MPCR    % OTHERWISE, CLEAR THE OVERFLOW BIT   01803000 D
                                                         %                                    01804000 D
                                                         %                                    01805000 D
0506  4809 A001 C030 00F0   CLEAROV: A1 C = A1                % THIS ROUTINE WILL CLEAR THE OVERFLOW 01806000 D
0507  3000 0000 0020 0030          27 = SAR              % BIT IN SR#1                         01807000 D
0508  4809 A00E 4000 40F0          A1 AND B11C = A1,CSAR                                      01808000 D
0509  2809 A001 C030 00F0          A1 C = A1; IF LC1     % CLEAR LC1 FROM CHECKOV              01809000 D
050A  482D 0000 0070 00F0          JUMP                                                       01810000 D
                                                         %                                    01811000 D
                                                         %                                    01812000 D
                                   SETOVBIT:             % THIS ROUTINE SETS THE OVERFLOW BIT IN 01813000 D
                                                         % POSITION 27 OF PSW                 01814000 D
050B  4809 A001 CC30 00F0          A1 C = A1                                                  01815000 D
050C  3000 0000 0020 C03C          27 = SAR                                                   01816000 D
050D  4809 ACDC 4030 40F0          A1 OR 1 = A1,CSAR                                          01817000 D
050E  4809 A001 CC00 00F0          A1 C = A1                                                  01818000 D
050F  482D 0000 0030 00F0          JUMP                                                       01819000 D
                                                         %                                    01820000 D
                                   % * * * * * * * * * * * END OF OV BIT SETTING* * * * * * * * * * * * 01821000 D
                                                         %                                    01822000 D
                                   PRTERR:               % THIS ROUTINE IS CALLED WHENEVER AN  01823000 D
                                                         % ERROR PRINTOUT IS GENERATED.       01824000 D
                                                         % B HAS THE ERRORLIST NUMBER.        01825000 D
0510  1F40 0000 0030 0060          CLEARBUFF - 1 = CPCR                                       01826000 D
0511  0A7C 0000 C000 0060          WRITEBUFF - 1 = CPCR  % WRITE A BLANK LINE                 01827000 D
0512  4809 0640 1030 0CF0          AMPCR = A3            % WRITE PRINTBUFF ADDRESS INTO A3     01828000 D
0513  0990 0000 0030 00C0          PRINTBUFF = AMPCR                                          01829000 D
0514  4809 2001 0800 00F0          LIT L = B             % LOAD COUNT IN UHW OF A3             01830000 D
0515  0080 0000 0000 00A0          8 = LIT; COMP 16 = SAR  % 8 WORDS WILL BE MOVED            01831000 D
0516  4809 EC5C 1000 00F0          A3 OR B = A3                                               01832000 D
0517  4A2C 0000 0030 0060          MOVE - 1 = CPCR       % MOVE ERROR DIAGNOSTIC INTO PRINTBUFF 01833000 D
0518  0A80 0000 0000 0060          WRITEBUFF - 1 = CPCR  % WRITE ERROR LIST MESSAGE           01834000 D
0519  1F40 0000 0000 0060          CLEARBUFF - 1 = CPCR                                       01835000 D
051A  0A90 0000 0000 0060          WRITEBUFF - 1 = CPCR                                       01836000 D
051B  0AAC C000 0070 0060          RESREG - 1 = CPCR     % RESTORE LV REGISTERS               01837000 D
051C  0AB0 0000 0000 0040          STOPTIMF - 1 = MFCR                                        01838000 D
                                                         %                                    01839000 D
                                                         %                                    01840000 D
                                   REGSTACK:             % THIS ROUTINE CHECKS THE STACK       01841000 D
                                                         % DESIGNATION BIT IN SR#1 TO DETERMINE 01842000 D
                                                         % WHICH STACK THE PROGRAMMER IS USING. 01843000 D
                                                         % THE B REGISTER IS ASSUMED TO HAVE THE 01844000 D
                                                         % RELATIVE REGISTER NUMBER.           01845000 D
                                                         % WRAPAROUND IS ALLOWED.              01846000 D
051D  4809 0C43 001C 00F0          B = MAR2              % STORE THE REGISTER                  01847000 D
051E  4809 A001 C00C C0F0          A1 C = A1             % POSITION STACK DESIGNATOR BIT IN LS BIT 01848000 D
051F  0000 0000 0020 0020          16 = SAR              % POSITION REG BIT. IN LS BIT         01849000 D
0520  4809 A000 0000 00F0          A1                    % SEND A1 TO THE ADDER                01850000 D
0521  580B 0FC3 0000 C0FC          IF LST THEN STEP ELSE SKIP                                 01851000 D
0522  0ACC 0003 0000 C040          STACK2 - 1 = MPCR                                          01852000 D
0523  4809 2C52 0000 00F0          LIT EOL D                                                  01853000 D
0524  010C 0000 0030 00EC          16 = LIT                                                   01854000 D
0525  580B 0000 000C 00F0          IF TRUE THEN STEP ELSE SKIP                                01855000 D
0526  4809 0C03 001C C0F0          0 = MAR2                                                   01856000 D
0527  482D A001 CC3C 00F0          A1 C = A1; JUMP       % RESTORE A1 TO NORMAL SR#1/PAR FORMAT 01857000 D
```

```
0528  4809 2C43 001C 00FC  STACK2I  LIT + B = MAR2                                  01858C00 0
0529  0100 0000 0F00 00E0           16 = LIT                                       01859000 0
052A  4809 2C52 0000 00F0           LIT EQL B                                      01860000 0
052B  0200 0000 0C90 C0E0           32 = LIT                                       01861CC0 0
052C  6819 0003 0C90 00F0           IF TRUE THEN SKIP                              01862C00 D
052D  4820 A0C1 C000 00F0           A1 C = A1 , JUMP                               01863C00 D
052E  4809 20C3 001C 00F0           LIT = MAR2                                     01864C00 C
052F  0100 0000 0000 00E0           16 = LIT                                       01865C0C D
0530  482D ACC1 C000 00F0           A1 C = A1, JUMP                                01866C00 0
                                                                  %                01867C00 D
                                                                  %                01868C0C 0
                          RESREG:                                 % THIS ROUTINE RESTORES THE REGISTERS  01869C00 C
                                                                  % OF THE LU IN WORKSPACE               01870C00 0
0531  4809 0640 0090 C0F0           AMPCR = MIR                   % SAVE RETURN IN MIR                   01871C00 D
0532  4809 20C3 001C C0F0           LIT + 1 = MAR2                % MAR2 = WORKSPACE + 1                  01872C00 0
0533  026C 0000 0000 00E0           WORKSPACE = LIT                                                      01873C00 D
0534  2F30 C003 0C00 006C           EINPUT - 1 = CPCR                                                    01874C00 D
0535  4809 0C40 4C30 00F0           B = A1                        % RESTORE A1                           01875C00 D
0536  4809 0F46 0C1C C0F0           BMAR + 1 = MAR2               % INCREMENT WORKSPACE ADDR             01876C00 0
0537  2F3C 0003 0C90 0060           EINPUT - 1 = CPCR                                                    01877C00 0
0538  4809 0C40 2F00 C0F0           B = A2                        % RESTORE A2                           01878C00 0
0539  4809 0F46 001C 00FC           BMAR + 1 = MAR2               % INCREMENT WORKSPACE ADDR             01879C00 D
053A  2F30 0003 0C0C C060           EINPUT - 1 = CPCR                                                    01880C00 D
053B  4809 0C43 1F00 00F0           B = A3                        % RESTORE A3                           01881F00 D
053C  4809 20C3 001C 00F0           LIT = MAR2                    % SELECT WORKSPACE BASE ADDR           01882CC0 0
053D  2F30 0003 0F00 C06C           EINPUT - 1 = CPCR                                                    01883CC0 C
053E  4809 0C40 0D90 C0F0           B = MIR,BMI                   % SWAP B AND MIR                       01884C00 0
053F  4809 0C40 0040 00F0           B = AMPCR,BMI                 % RESTORE AMPCR AND B                   01885CC0 D
0540  4809 0003 0C90 C0F0           STEP                                                                 01886C0C D
0541  482D 0003 0000 00F0           JUMP                                                                 01887000 D
                                                                  %                01888C00 0
                                                                  %                01889C00 C
                          RESTPSW:                                % THIS ROUTINE WILL PLACE THE CONTENTS  01890C06 0
                                                                  % OF PSW INTO THE PAR, THEN RESTORE    01891C00 D
                                                                  % THE PAR TO (PAR) + 2.                01892C00 0
0542  4809 0640 0090 0CFC           AMPCR = MIR                   % SAVE RETURN ADDR TO OPCODE IN MIR     01893C00 D
0543  4809 20C3 001C C0F0           LIT = MAR2                    % ADDRESS OF PSW                        01894C00 D
0544  0200 0003 0000 C0E0           PSW = LIT                                                            01895C00 D
0545  2F3C 00C0 0F00 C060           EINPUT - 1 = CPCR             % (PSW) INTO B                         01896C00 0
0546  4809 0C40 0D90 C0F0           B = MIR,BMI                   % (PSW) INTO MIR, RETURN ADDR INTO B   01897C00 0
0547  4809 A0C1 2000 00F0           A1 L = A2                     % PAR INTO A2                          01898CC0 D
0548  0020 0003 0000 00A0           COMP 16 = SAR, 2 = LIT                                               01899000 D
0549  4809 CCC0 A030 00F0           A2 R = A2                     % RIGHT JUSTIFY (PAR)                  0190CC00 D
054A  4809 C141 2000 C0F0           A2 + LIT L = A2               % (PAR) + 2 = A2                       0191C1C0 D
054B  4809 CC5C 1000 00F0           A2 OR B = A3                  % (PAR)+2/RETURN ADDR = A3             01902C00 0
054C  4809 A0C3 C000 00F0           A1 R = A1                     % CLEAR PAR                            01903C00 0
054D  4809 A0C1 4030 00FC           A1 L = A1, BMI                                                       0190400C 0
054E  4809 AC5C 4030 00F0           A1 OR B = A1                  % (PSW) INTO PAR                       01905C00 D
054F  4809 B80E 4C00 CCF0           A1 AND B011 = A1              % CLEAR BIASED FETCH BIT               01906C00 D
0550  0A0C 0003 0C90 C060           IFETCH - 1 = CPCR             % INSTRUCTION AT Y INTO B              01907F00 D
0551  4809 0C40 0C90 00F0           B = MIR                                                              01908C00 D
0552  4809 A000 C000 00F0           A1 R = A1                     % CLEAR PAR                            01909000 0
0553  0000 0003 0F30 0020           16 = SAR                                                             0191C000 0
0554  4809 A0C1 4030 C0F0           A1 L = A1                                                            01911C00 D
0555  4809 E003 8F90 00F0           A3 R = B                      % (PAR) + 2 = F                        01912C0C D
0556  4809 AC5C 4000 00F0           A1 OR B = A1,BMI                                                     01913C0C 0
0557  4809 E000 0030 C0F0           A3 = AMPCR                    % RESTORE RETURN ADDR, INSTR IN B       01914C00 0
0558  4809 00C0 0000 00F0           STEP                                                                 01915C0C D
0559  483F 0000 0000 00F0           RETN                                                                 01916C00 D
                                                                  %                01917C00 D
```

```
                                     *
                             RI1:    * THIS ROUTINE COMPUTES (P)+ D = P
                                     * AND JUMPS TO NEXT INSTRUCTION
055A  4809 0000 0B00 00F0         C = B
055B  4809 EC01 100C 40F0         A3 L = A3, CSAR         * EXAMINE SIGN BIT OF "D"
055C  0000 0000 000C C010         COMP 24 = SAR
055D  4809 E003 0030 C0FC         A3                      * A3 TO THE ADDER
055E  4C09 0019 CB0C 00F0         IF MST THEN 0111 L = B  * IF SIGN=1, FILL UPPER BYTE WITH 1:S
055F  4809 E000 9000 C0F0         A3 R = A3               * RESTORE "D" FIELD TO LS BYTE
0560  48C9 EC53 0BC0 00F0         A3 OR B L = B           * B = SIGN/"D", IN MS WORD
0561  0000 0000 0000 C020         COMP 16 = SAR
0562  4809 ADC1 1000 C0F0         A1 L = A3               * SAVE THE OLD PAR IN MS WORD OF A3
0563  4809 EC40 8B00 C0F0         A3 + B R = B            * ALGEBRAICALLY ADD (P) + D
0564  4809 AC03 C000 00F0         A1 R = A1               * CLEAR OLD PAR
0565  4809 A001 400C 00FC         A1 L = A1
0566  4809 AC5C 4000 00F0         A1 OR B = A1            * CREATE NEW PAR
0567  0AEC 0000 0030 0040         OPCODE - 1 = MPCR
                                     *
                                     *
                             RM:     * THIS ROUTINE GETS THE CONTENTS OF R(M)
                                     * :M: INDICATED IN MAR AND RETURNS
                                     * IT IN MIR
0568  4809 0000 000C 24F0         ASE                     * REFERENCE MAR2
0569  4809 0640 2000 C0F0         AMPCR = A2              * SAVE RETURN ADDRESS
056A  4809 0F40 0B0C 00F0         BMAR = B
056B  51C0 0003 0000 0060         REGSTACK - 1 = CPCR     * SELECT REGISTER STACK TO BE USED
                                                          * ADDR OF GEN REG STACK RETURNED IN MAR2
056C  2F30 0000 0000 0060         EINPUT - 1 = CPCR       * GET CONTENTS OF R(M)
056D  48C9 CCC0 004C C0F0         A2 = AMPCR              * RESTORE RETURN ADDRESS
056E  4809 0C40 0C3C 00F0         B = MIR                 * MIR IS USED AS TEMP STORAGE
056F  4800 0000 000C 00F0         WAIT                    * FOR TEST PURPOSES
0570  482D 0000 0030 00F0         JUMP
                                     *
                                     *
* * * * * * * * * * * * * RX FORMAT ROUTINES * * * * * * * * * * * * * *
                                     *
                                     *
                             RXMFIELD:  * THIS ROUTINE ANALYZES THE "M" FIELD OF
                                     * A TYPE RX INSTRUCTION.  UPON ENTRANCE
                                     * A3 CONTAINS THE MOST SIGNIF. WORD OF
                                     * THE DOUBLEWORD INSTRUCTION.  LC2
                                     * INDICATES BYTE INSTRUCTION.  UPON EXIT
                                     * OF DAWI, DAWDI, OR INDIRECTION
                                     * B CONTAINS Y: LC2 PASSES MS(0),LS(1)
0571  4809 E001 9000 00F0         A3 C = A3               * PREPARE MS 2 BYTES OF A3 TO HOLD RETURN
0572  000C 0000 000C 0020         16 = SAR                * ADDRESS
0573  48C9 E65C 1000 00F0         A3 OR AMPCR = A3        * PUT RETURN ADDR IN A3(UPPER HALF)
0574  4809 E001 9C3C 00FC         A3 C = A3               * RESTORE A3 RETURN ADDR/INSTRUCTION
0575  4809 E156 0B0C 00F0         A3 AND LIT = B          * MASK OFF "M" FIELD
0576  0CFC 0003 0030 00E0         15 = LIT
0577  4809 0C52 0C00 C0F0         B EQL 0                 * IS M = 0?
0578  680B 0000 0000 C0F0         IF TRUE THEN STEP ELSE SKIP
0579  0AFC 0000 000C 004C         DAWDI - 1 = MPCR        * JUMP TO DIRECT ADDRESS W/O INDEXING
057A  4809 2C5B 0C00 0CF0         B GEQ LIT               * TEST FOR M = 1XXX
057B  0080 0000 0000 C0E0         8 = LIT
057C  780B 0000 000C 00F0         IF FALSE THEN STEP ELSE SKIP
057D  0B0C 0000 0C3C 0040         DAWI - 1 = MPCR         * JUMP TO DIRECT ADDR WITH INDEXING
057E  4809 0C40 2030 C0F0         B = A2                  * SAVE B IN A2
057F  500B 0000 0000 00F0         IF NOT LST THEN STEP ELSE SKIP
0580  0B1C 0C03 C000 C040         INDIR - 1 = MPCR        * IF M = 10,12,14,16 THEN INDIRECTION
```

```
0581 0B2C 0000 0000 0040           DAW1 - 1 = MPCR        * OTHERWISE, M = 11,13,15,17 PERFORM    01978000 D
                                                          * DIRECT ADDRESSING WITH INDEXING (R(M)) 01979000 D
                                                          *                                       01980000 D
                                                          *                                       01981000 D
                         DAWOI:                           * THIS ROUTINE RETURNS Y IN D           01982000 D
                                                          * AND JUMPS TO CALLING ROUTINE          01983000 D
                                                          * OF RXMFIELD                           01984000 D
0582 0B30 0000 0000 0060           IFETCH - 1 = CPCR      * B HAS Y                               01985000 D
0583 4809 E000 8030 00F0           A3 R = AMPCR           * RESTORE RETURN ADDR FROM RXMFIELD     01986000 D
0584 0000 0000 0030 C020           16 = SAR                                                       01987000 D
0585 3809 0000 0000 00F0           IF LC2                 * CLEAR BYTE BIT, LC2=0; MSBYTE OF Y    01988000 D
0586 4820 0000 0000 00F0           JUMP                   * RETURN TO CALLING ROUTINE OF RXMFIELD 01989000 D
                                                          *                                       01990000 D
                                                          *                                       01991000 D
                         DAWI:                            * THIS ROUTINE FORMS Y = Y + (R(M))     01992000 D
                                                          * RETURNS Y IN B                        01993000 D
0587 0B40 0000 0030 0060           IFETCH - 1 = CPCR      * GET NEXT ADDRESS                      01994000 D
0588 4009 0C40 2030 00F0           B = A2                 * A2 HAS "Y"                            01995000 D
0589 4809 E155 0B00 00FC           A3 AND LIT = B         * RETRIEVE RELATIVE  REG. NO.           01996000 D
058A 00FC 0000 0000 00E0           15 = LIT               * MASK                                  01997000 D
058B 51C0 0000 0000 0060           REGSTACK - 1 = CPCR    * SELECT REG. STACK TO BE USED          01998000 D
                                                          * COMPUTED ACTUAL LOCATION IN MAR2      01999000 D
058C 2F3C 0000 0030 C060           EINPUT - 1 = CPCR      * B HAS (R(M))                          02000000 D
058D 3A9B 0000 0000 C0F0           IF LC2 THEN SET LC2; STEP ELSE SKIP                            02001000 D
058E 1E1C 0000 0030 0060           BYTETEST - 1 = CPCR    * CHECK FOR BYTE INSTRUCTION            02002000 D
058F 4809 CC40 0B00 00F0           A2 + B = B             * B HAS Y                               02003000 D
0590 4809 E000 8030 00F0           A3 R = AMPCR           * RESTORE RETURN ADDR FROM RXMFIELD     02004000 D
0591 0000 0000 0030 0020           16 = SAR                                                       02005000 D
0592 4009 0000 0000 00F0           STEP                                                           02006000 D
0593 4820 0000 0000 00F0           JUMP                   * RETURN TO CALLING ROUTINE OF RXMFIELD 02007000 D
                                                          *                                       02008000 D
                                                          *                                       02009000 D
                         INDIR:                           * THIS ROUTINE IS CALLED IF M=10,12,14,1602010000 D
0594 4809 2000 001C C0F0           LIT = MAR2             * LOAD ADDR OF SR#2 REGISTER INTO MAR2  02011000 D
0595 0220 0000 0030 00E0           STATUS2 = LIT                                                  02012000 D
0596 2F30 0000 0000 0060           EINPUT - 1 = CPCR      * READ SR#2 INTO B TO USE SRM ROUTINES  02013000 D
0597 4809 C152 0030 00F0           A2 EOL LIT                                                     02014000 D
0598 0080 0000 0000 00E0           8 = LIT                                                        02015000 D
0599 6419 C152 0000 C0F0           IF FALSE THEN A2 EOL LIT; SKIP  * TEST IF M = 10               02016000 D
059A 0B50 0000 0030 0040           SRM10 - 1 = MPCR       * JUMP TO SR#2 "M" FIELD ROTUINE        02017000 D
059B 00AC 0000 0030 00E0           10 = LIT                                                       02018000 D
059C 6419 C152 0000 C0F0           IF FALSE THEN A2 EOL LIT; SKIP  * TEST IF M = 12               02019000 D
059D 0B60 0000 000C 0040           SRM12 - 1 = MPCR                                               02020000 D
059E 00CC 0000 0030 00E0           12 = LIT                                                       02021000 D
059F 6019 C000 0030 00FC           IF FALSE THEN SKIP     * TEST IF M = 14                        02022000 D
05A0 0B7C 0000 0000 0040           SRM14 - 1 = MPCR                                               02023000 D
05A1 0B80 C000 0000 C040           SRM16 - 1 = MPCR       * DEFAULTS TO M = 16                    02024000 D
                                                          *                                       02025000 D
                                                          *                                       02026000 D
                         SRM10:                           * THIS ROUTINE WILL REMOVE THE "M" FIELD 02027000 D
                                                          * FROM SR#2 AND CALL THE APPROPRIATE    02028000 D
                                                          * INDIRECT ADDRESSING ROUTINE           02029000 D
05A2 4809 18C1 AC30 00F0           B101 C = A2            * ISOLATE :M: FIELD = 10                02030000 D
05A3 3000 0000 0030 0020           23 = SAR                                                       02031000 D
05A4 4809 CC56 8B30 C0F0           A2 AND B R = B         * B CONTAINS STATUS REG 2               02032000 D
05A5 0000 0000 0000 0010           8 = SAR                                                        02033000 D
05A6 0B90 0000 0030 C040           SRMTEST - 1 = MPCR     * JUMP TO TEST CONTENTS ROUTINE         02034000 D
                                                          *                                       02035000 D
                                                          *                                       02036000 D
                         SRM12:                           * THIS ROUTINE WILL REMOVE THE :M: FIELD 02037000 D
```

```
                                                    % INDIRECT ADDRESSING ROUTINE                    02038C00 D
05A7  4809 18C1 A000 00F0                 0101 C = A2           % ISOLATE :M: FIELD = 12             02039C00 D
05A8  9000 0000 0030 0020                 21 = SAR                                                  02040000 D
05A9  4809 CC56 8B30 00F0                 A2 AND 0 R = B        % B CONTAINS STATUS REG 2            02041C00 D
05AA  2000 00C0 0000 0010                 10= SAR               % P CONTAINS CONTENTS OF :M: FIELD   02042C00 D
05AB  0BA0 00C0 0030 0040                 SRMTEST - 1 = MPCR    % JUMP TO TEST CONTENTS ROUTINE      02043C00 D
                                                    %                                               02044C00 D
                                                    %                                               02045C00 D
                                          SRM14:                % THIS ROUTINE WILL REMOVE THE :M: FIELD 02046C00 D
                                                    % FROM SR#2 AND CALL THE APPROPRIATE             02047C00 D
                                                    % INDIRECT ADDRESSING ROUTINE                    02048C00 D
05AC  4809 18C1 A030 00F0                 0101 C = A2           % ISOLATE :M: FIELD = 14             02049C00 D
05AD  3000 00C0 C030 0020                 19 = SAR                                                  02050C00 D
05AE  4809 CC56 8B00 00F0                 A2 AND B R = B        % B CONTAINS STATUS REG 2            02051C00 D
05AF  8000 0C00 0030 0010                 12= SAR               % R CONTAINS CONTENTS OF :M: FIELD   02052C00 D
05B0  0B0C 00C0 0030 0040                 SRMTEST - 1 = MPCR    % JUMP TO TEST CONTENTS ROUTINE      02053C00 D
                                                    %                                               02054C00 D
                                                    %                                               02055C00 D
                                          SRM16:                % THIS ROUTINE WILL REMOVE THE :M: FIELD 02056C00 D
                                                    % FROM SR#2 AND CALL THE APPROPRIATE             02057C00 D
                                                    % INDIRECT ADDRESSING ROUTINE                    02058C00 D
05B1  4809 18C1 A030 00F0                 0101 C = A2           % ISOLATE :M: FIELD = 16             02059C00 D
05B2  1000 00C0 0030 0020                 17 = SAR                                                  02060C00 D
05B3  4809 CC56 8B30 00F0                 A2 AND B R = B        % B CONTAINS STATUS REG 2            02061C00 D
05B4  A00C 0CC0 0C0C 0010                 14= SAR               % B CONTAINS CONTENTS OF :M: FIELD   02062C00 D
05B5  0BC0 C000 0030 0040                 SRMTEST - 1 = MPCR    % JUMP TO TEST CONTENTS ROUTINE      02063C00 D
                                                    %                                               02064C00 D
                                                    %                                               02065C00 D
                                          SRMTEST:              %  THIS ROUTINE ANALYSES THE :M: FIELD 02066C00 D
                                                    % CONTENTS M = 0,1,2,3                           02067C00 D
05B6  4809 2C5E 0030 00F0                 LIT GEQ B                                                 02068C00 D
05B7  0010 00C0 0000 00E0                 1 = LIT                                                   02069C00 D
05B8  7419 2C52 0000 00F0                 IF FALSE THEN LIT EQL B; SKIP  % TEST CONTENTS OF "M"(0 OR 1) 02070C00 D
05B9  586C 0003 0C00 C040                 DAWI - 1 = MPCR       % DIRECT ADDRESSING WITH INDEXING   02071C00 D
05BA  0020 0000 0C00 C0E0                 2 = LIT                                                   02072C00 D
05BB  6819 00C0 C030 00F0                 IF TRUE THEN SKIP     % TEST IF "M" = 2                   02073C00 D
05BC  401C 0000 0C0C C040                 IAWI - 1 = MPCR       % JUMP TO INDIRECT ADDR WITH INDEXING 02074000 D
05BD  4100 0000 C030 0040                 IAWOI - 1 = MPCR      % INDIRECT ADDR WITHOUT INDEXING     02075C00 D
                                                    %                                               02076C00 D
                                          % * * * * * * * * * * END RX FORMAT ROUTINES * * * * * * * * * * * * * * * * 02077C00 D
                                                    %                                               02078000 D
                                          SAVEREG:              % THIS ROUTINE SAVES LU(LOGIC UNIT) REG. 02079C00 D
05BE  4809 0003 0C0C 24F0                 ASE                   % REFERENCE BR2/MAR                  02080C00 D
05BF  4809 0C40 00B0 00FC                 B = MIR               % STORE B IN MIR                     02081C00 D
05C0  4809 0640 0B0C 00F0                 AMPCR = B             % SAVE RETURN IN B                   02082C00 D
05C1  4809 2003 0C1C C0F0                 LIT = MAR2                                                02083C00 D
05C2  026C 0C0D 0C3C C0E0                 WORKSPACE = LIT       % WRITE B INTO WORKSPACE             02084C00 D
05C3  2F50 C0C3 0C00 0060                 EOUTPUT - 1 = CPCR                                        02085C00 D
05C4  4809 0F46 0C1C 00F0                 BMAR + 1 = MAR2                                           02086C00 D
05C5  4809 A000 0030 C0F0                 A1 = MIR              % WRITE A1 INTO WORKSPACE +1         02087CC0 D
05C6  2F50 00C0 0000 0060                 EOUTPUT - 1 = CPCR                                        02088C00 D
05C7  4809 0F46 0C1C 00F0                 BMAR + 1 = MAR2                                           02089C00 D
05C8  4809 C0D0 0C30 C0F0                 A2 = MIR              % WRITE A2 INTO WORKSPACE + 2        0209CC00 D
05C9  2F50 00C3 0C00 0060                 EOUTPUT - 1 = CPCR                                        02091C00 D
05CA  4809 0F46 0C1C 00F0                 BMAR + 1 = MAR2                                           02092C00 D
05CB  4809 E000 0C30 00F0                 A3 = MIR              % WRITE A3 INTO WORKSPACE + 3        02093C00 D
05CC  2F50 00C3 0C0C 0060                 EOUTPUT - 1 = CPCR                                        02094C00 D
05CD  4809 0C40 B04C 00F0                 B = AMPCR             % RESTORE ADDRESS                    02095C00 D
05CE  4809 00C3 0000 00FC                 STEP                                                      02096C00 D
                                                                                                    02097C00 D
```

```
05CF  4820 C000 0000 00F0          JUMP                                                    02098000 0
                                                              *                            02099000 0
                                                              *                            02100000 0
                         STARTTIME:                           * THIS ROUTINE STARTS THE EXTERNAL      02101000 0
                                                              * CLOCK OF THE IOP.                     02102000 0
05D0  4809 2001 0B00 00F0          LIT L = B                                               02103000 0
05D1  0020 0000 0000 C0A0          2 = LIT; COMP 16 = SAR     * PUT FUNCTION 2 INTO B      02104000 0
05D2  4809 2050 0030 00FC          LIT OR B = MIR             * PUT START CLOCK FUNCTION INTO MIR     02105000 0
05D3  0010 0000 0000 00E0          1 = LIT                                                 02106000 0
05D4  2F70 0000 0000 0060          EXTIO - 1 = CPCR           * CALL IOP                   02107000 0
05D5  4809 0000 0000 00FC          STEP                                                    02108000 0
05D6  0B00 0000 0000 0040          OPCODE - 1 = MPCR          * START EXECUTION            02109000 0
                                                              *                            02110000 0
                                                              *                            02111000 0
                         STOPTIME:                            * THIS ROUTINE SAMPLES THE EXTERNAL     02112000 0
                                                              * CLOCK OF THE IOP AND INSIRTS THE TIME 02113000 0
                                                              * INTO CLOCKTIME IN S-MEMORY AND CALLS  02114000 0
                                                              * THE LOADER.                           02115000 0
05D7  4809 2001 0030 00F0          LIT L = MIR                * STORE FUNCTION IN MIR      02116000 0
05D8  0020 0000 0000 C0A0          2 = LIT; COMP 16 = SAR     * PUT FUNCTION 2 IN B        02117000 0
05D9  2F70 0000 0000 0060          EXTIO - 1 = CPCR           * CALL IOP                   02118000 0
05DA  5060 0000 0000 004C          STOPTIME - 1 = MPCR        * REREAD TIME                02119000 0
05DB  4809 0C40 0030 00FC          B = MIR                    * PUT CONTENTS OF MAILBOX INTO MIR      02120000 0
05DC  4809 2000 001C 00F0          LIT = MAR2                                              02121000 0
05DD  024C 0000 0000 00E0          CLOCKTIME = LIT            * CLOCKTIME ADDRESS IN MAR2  02122000 0
05DE  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR         * WRITE CLOCK INTO CLOCKTIME 02123000 0
05DF  4809 2000 001C 00F0          LIT = MAR2                                              02124000 0
05E0  0220 0000 0000 00EC          STATUS2 = LIT                                           02125000 0
05E1  2F30 0000 0000 0060          EINPUT - 1 = CPCR          * READ IN SR #2              02126000 0
05E2  4809 0C40 0000 00FC          B                          * SR # 2 TO THE ADDER        02127000 0
05E3  4806 0000 0000 00F0          IF MST THEN STEP ELSE SKIP  * TRACE IF TRUE             02128000 0
05E4  27E0 0000 0000 0060          DUMPREG - 1 = CPCR         * DUMP AN/UYK20 REGISTERS, ETC          02129000 0
05E5  0600 0000 0000 0040          NEWCARD - 1 = MPCR         * RETURN TO LOADER FOR ADDITIONAL INPUT 02130000 0
                                                              *                            02131000 0
                                                              *                            02132000 0
                         WRITEBUFF:                           * THIS ROUTINE WILL PRINTOUT THE CONTENTS02133000 0
                                                              * OF PRINTBUFF TO THE CRT OR PTR        02134000 0
                                                              * A2, MIR, AND B ARE USED IN THIS ROUTINE02135000 0
05E6  4809 0640 2000 00F0          AMPCR = A2                 * SAVE AMPCR                 02136000 0
05E7  4809 2000 0030 C0F0  BDOUT:  LIT = MIR                  * PUT ADDRESS OF PRINTBUFF IN MIR       02137000 0
05E8  0990 0000 0000 00E0          PRINTBUFF = LIT                                         02138000 0
05E9  4809 2000 001C C0FC          LIT = MAR2                 * ADDR OF SR#2 INTO MAR2     02139000 0
05EA  B220 0000 0000 00B0          STATUS2 = LIT; COMP 1 = SAR                             02140000 0
05EB  2F30 0000 0000 0060          EINPUT - 1 = CPCR          * SR#2 INTO B                02141000 0
05EC  4809 0C41 0B00 00FC          B L = B                    * EXAMINE 2ND MS BIT OF SR#2 02142000 0
05ED  4809 0000 0000 00FD          STEP                                                    02143000 0
05EE  007C 0000 0000 00AC          COMP 16 = SAR; 7 = LIT                                  02144000 0
05EF  4809 0C40 0000 00F0          B                          * BRING B TO THE ADDER       02145000 0
05F0  4C19 2001 0F00 00F0          IF MST THEN LIT L = BBI; SKIP  * CRT FUNCTION           02146000 0
05F1  4809 0001 0F00 C0FC          1 L = BBI                  * PRINTER FUNCTION           02147000 0
05F2  4809 0C40 0030 00FC          B = MIR                    * MIR CONTAINS 1/PRINTBUFF OR 7/PRINTBUFF02148000 0
05F3  4809 2000 0B00 00F0          LIT = B                    * 33 INTO B                  02149000 0
05F4  021C 0000 0000 00E0          33 = LIT                                                02150000 0
05F5  2F70 0000 0000 0060          EXTIO - 1 = CPCR           * WRITE TO CRT/PTR 33 WORDS IN PBUFF    02151000 0
05F6  5E60 0000 0000 004C          BDOUT - 1 = MPCR           * REPEAT EXTIO, I/O NOT COMPLETE        02152000 0
05F7  4809 C000 0040 C0FD          A2 = AMPCR                 * RESTORE AMPCR              02153000 0
05F8  4809 0000 0000 00FD          STEP                       * REQUIRED BETWEEN 2 TYPE II INSTRUCTIONS02154000 0
05F9  4820 0000 0000 00F0          JUMP                                                    02155000 0
                                                              *                            02156000 0
                                                              *                            02157000 0
```

```
                                XFCODE:                 % THIS ROUTINE WILL EXAMINE THE "F" CODE 02158C00 D
05FA  4809 E0C0 A0DC 00F0               A3 R = A2       % OF AN INSTRUCTION IN B AND RETURN THE  02159C00 D
05FB  0000 0000 0000 C010               B = SAR         % "F" CODE IN LS 2 BITS OF A2            0216CC00 D
05FC  4809 C155 2030 00F0               A2 AND LIT = A2                                          02161C00 D
05FD  0030 C000 0000 00EC               3 = LIT                                                  02162C00 D
05FE  482D 0000 0000 00F0               JUMP                                                     02163C00 D
                                        %                                                        02164000 D
                                % * * * * * * * * * * * * * * FETCH ROUTINES * * * * * * * * * * * * * * * *02165C00 D
                                        %                                                        02166000 D
                                BKTCMP:                 % BKPT STATUS BIT HAS BEEN SET IN SR#1   02167C00 D
                                                        % BR1 CONTAINS THE ADDRESS TO BE MATCHED.02168C00 D
                                                        % BR2 CONTAINS THE DIRECT ADDRESS        02169C00 D
                                                        % A1,A2,A3,AMPCR MUST BE SAVED TO BE USED02170C00 D
05FF  4809 C000 0030 00FC               A2 = MIR        % SAVE A2 IN MIR                          02171C00 D
0600  4809 00C0 0000 24F0               ASE             % REFERENCE BR2/MAR                      02172C00 D
0601  4809 0F40 A030 00FC               BMAR R = A2     % SAVE BR2 IN A2                         02173C00 D
0602  0210 C000 0030 0090               B = SAR; BKPT = LIT                                      02174C00 D
0603  48C9 2003 C01C 00F0               LIT = MAR2      % BREAKPOINT LOCATION IN MAR2            02175C00 D
0604  A8C9 0003 0000 0CF0               MR2; IF RDC     % PERFORM READ OF BKPT                   02176C00 D
0605  AC08 00C3 0C70 00FC               WHEN RDC THEN BEX % B CONTAINS BREAKPOINT                02177000 D
0606  4809 CC52 0C00 C0F0               A2 EOL B        % ER2 EQUAL BKPT                         02178C00 D
0607  68C8 0003 0030 00FC               IF TRUE THEN STEP ELSE SKIP  %IF ADDRESSES MATCH STOP EXECUTIONO2179000 D
0608  4800 0000 0C30 00FC               WAIT                                                     0218CC00 D
0609  4809 C0C1 001C 00F0               A2 L = BR2      % RESTORE ER2                            02181C00 D
060A  0000 0003 0030 0030               COMP B = SAR                                             02182C00 D
060B  4809 0000 0D00 00F0               BMI             % RESTORE A2 TO ORIG. CONTENTS           02183C0C D
060C  2808 0C40 2C30 00F0               B = A2; IF LC1 THEN STEP ELSE SKIP % CALLING ROUTINE CHECK 02184C00 D
060D  0BEC 0000 0000 C04C               BKPTOUT - 1 = MPCR % RETURN TO EMULOUT                   02185000 D
060E  0BFC 0000 0C00 C04C               BKPTIN  - 1 = MPCR % RETURN TO EMULIN                    02186000 D
                                        %                                                        02187CC0 D
                                        %                                                        02188000 D
                                EMULIN:                 % THIS ROUTINE READS FROM MEMORY AFTER   02189C0C D
                                                        % CHECKING THE PAGING AND FREAKPOINT BITS02190C00 D
                                                        % IN SR#1.  BR2 CONTAINS THE ADDRESS TO  02191C00 D
                                                        % BE READ FROM.                          02192000 D
060F  2809 A0C0 8B30 00F0               A1 R = B; IF LC1                                         02193C00 D
0610  A00C 0003 0070 0020               22 = SAR        %                                        02194C00 D
0611  4809 0C40 0030 00F0               B               % SEND B TO THE ADDER                    02195C00 D
0612  5808 00C3 0030 00FC               IF LST THEN STEP ELSE SKIP  % TEST PAGING BIT            02196C00 D
0613  0C00 00C3 003C 004C               PAGING - 1 = MPCR % JUMP IF PAGING BIT SET               02197CC0 D
0614  4809 ACC0 880C 00F0     PAGEIN:   A1 R = B                                                 02198C00 D
0615  800C 0003 C030 0020               20 = SAR                                                 02199000 D
0616  4809 CC40 0030 C0F0               B               % SEND B TO THE ADDER                    02200C00 D
0617  580B 00C3 0000 00F0               IF LST THEN STEP ELSE SKIP  % TEST BREAKPOINT BIT        02201C00 D
0618  5FEC 0000 0000 004C               BKTCMP - 1 = MPCR % JUMP TO EKPT ANALYSIS ROUTINE        02202C0C D
0619  4809 0000 0C00 24F0     BKPTIN:   ASE             % REFFRENCE MAR2                         02203000 D
061A  4809 0F40 801C 00F0               BMAR R = MAR2   % SET UP MAR2 WITH ADDRESS               02204000 D
061B  0020 0003 0030 0010               B = SAR                                                  02205000 D
061C  A809 0003 0C00 0CF0               MR2; BEX; IF RDC  % READ IN PRESENT ADDRESS INTO B       02206000 D
061D  AC08 0003 0000 00F0               WHEN RDC THEN 0                                          02207C00 D
061E  482D 00C3 0C30 00F0               JUMP                                                     02208000 D
                                        %                                                        02209C00 D
                                        %                                                        0221CC00 D
                                EMULOUT:                % THIS ROUTINE WRITES OUT INTO MEMORY     02211C00 D
                                                        % AFTER CHECKING THE PAGING BIT(BIT 22)  02212C00 D
                                                        % AND THE BREAKPOINT BITS(BITS 21-20) IN 02213C00 D
                                                        % THE SR#1 PORTION OF THE PSW.  BR2      02214C00 D
                                                        % CONTAINS THE ADDRESS TO BE WRITTEN      02215C00 D
                                                        % INTO AND MIR CONTAINS THE INFO TO BE   02216C00 D
                                                        % WRITTEN.                               02217C00 D
```

```
061F  4909 0000 0030 00F0          SET LC1                  % FLAG USED TO IDENTIFY THIS ROUTINE     02218000 D
0620  4809 0000 0030 00F0          BMI                      % TRANSFER MIR TO B                      02219000 D
0621  4809 CC40 2000 00F0          0 = A2                   % TRANSFER B(MIR) TO A2                  02220000 D
0622  4809 A000 8B00 00F0          A1 R = B                                                          02221000 D
0623  A00C 000C 0000 C020          22 = SAR                                                          02222000 D
0624  4809 CC40 0030 00F0          B                        % SEND B TO THE ADDER                    02223000 D
0625  580B 0000 0000 C0F0          IF LST THEN STEP ELSE SKIP  % CHECK IF PAGING SELECTED            02224000 D
0626  9C1C 000C 0C30 004C          PAGING - 1 = MPCR        % JUMP TO PAGING ROUTINE                 02225000 D
0627  4809 A000 8B00 C0FC    PAGEOUT: A1 R = B                                                       02226000 D
0628  900C 000C 0030 0020          21 = SAR                                                          02227000 D
0629  4809 0C40 0030 00F0          B                        % TEST IF BREAKPOINT SELECTED            02228000 D
062A  5806 000C 0030 C0F0          IF LST THEN STEP ELSE SKIP                                        02229000 D
062B  5FFC C003 0B00 C040          BKTCMP - 1 = MPCR        % JUMP TO BKPT ANALYSIS ROUTINE          02230000 D
062C  4809 000C 0000 24F0    BKPTOUT: ASE                   % REFERENCE MAR2                         02231000 D
062D  4809 0F43 801C 00FC          BMAR R = MAR2            % MAR2 CONTAINS ADDRESS                  02232000 D
062E  0000 0000 C03C 0010          8 = SAR                                                           02233000 D
062F  4809 C000 0030 00F0          A2 = MIR                 % RESTORE MIR                            02234000 D
0630  9809 000C 0C30 1CF0          MW2, IF SAI              % PERFORM WRITE INTO MEMORY              02235000 D
0631  9C08 000C C030 00F0          WHEN SAI THEN C/ STEP                                             02236000 D
0632  2809 0C00 0030 00FC          IF LC1                   % CLEAR LC1                              02237000 D
0633  482D 0000 003C 00FC          JUMP                                                              02238000 D
                                                           %                                        02239000 D
                                                           %                                        02240000 D
                               IFETCH:                     % THIS ROUTINE WILL FETCH AN INSTRUCTION  02241000 D
                                                           % AND ENABLE A TRACE OF ALL REGISTERS IF  02242000 D
                                                           % PRESELECTED VIA A TRACE CONTROL CARD    02243000 D
0634  4809 0640 203C 00F0          AMPCR = A2               % STORE RETURN ADDRESS                   02244000 D
0635  4809 2003 0C1C 00FC          LIT = MAR2                                                        02245000 D
0636  022C 000C 0C3C 00E0          STATUS2 = LIT                                                     02246000 D
0637  2F30 000C 0C30 0060          EINPUT - 1 = CPCR        % READ IN SR #2                          02247000 D
0638  4809 0C40 0000 00FC          B                        % SR # 2 TO THE ADDER                    02248000 D
0639  4800 000C 0C00 00F0          IF MST THEN STEP ELSE SKIP  % TRACE IF TRUE                       02249000 D
063A  27EC 000C 0C3C 0060          DUMPREG - 1 = CPCR       %JUMP TO REGISTER DUMPING ROUTINE        02250000 D
063B  4809 A001 0B3C 00FC          A1 L = B                 % ISOLATE ADDRESS IN PAR FIELD OF PSW     02251000 D
063C  0000 000C 0030 0020          COMP 16 = SAR                                                     02252000 D
063D  4809 0C40 8B0C 00F0          B R = B                                                           02253000 D
063E  4809 0C41 0010 00FC          B L = BR2                % INSERT ADDRESS INTO BR2                 02254000 D
063F  000C C0C3 0C30 C030          COMP 8 = SAR             % BR2 IS FILLED FROM 2ND LSB OF SOURCE   02255000 D
0640  60E0 CC00 C000 C060          EMULIN - 1 = CPCR        % RETRIEVE INSTRUCTION AT ADDRESS IN BR2 02256000 D
0641  4809 C0C0 003C C0FC          A2 = AMPCR               % RESTORE RETURN ADDRESS                  02257000 D
0642  4809 A001 2000 C0F0          A1 L = A2                % ISOLATE PAR INTO A2                     02258000 D
0643  0000 00C0 003C C020          COMP 16 = SAR                                                     02259000 D
0644  4809 C000 A03C 00F0          A2 R = A2                                                         02260000 D
0645  4809 C0C0 A000 00F0          A2 + 1 R = A2            % TEST FOR UPPER MEMORY BOUNDRY          02261000 D
0646  4869 C018 0030 C0F0          A2 GTR C                                                          02262000 D
0647  7429 A0C0 4000 C0F0          IF FALSE THEN A1 + 1 = A1, JUMP  % NORMAL INCREMENT               02263000 D
0648  4809 0C40 0030 00F0          B = MIR                                                           02264000 D
0649  4809 00C1 0B0C 00F0          1 L = B                  % CREATE 1024 IN B                       02265000 D
064A  A000 0000 0000 C020          COMP 10 = SAR                                                     02266000 D
064B  4809 A0C0 C000 C0FC          A1 R = A1                                                         02267000 D
064C  00C0 0000 003C 002C          16 = SAR                                                          02268000 D
064D  4809 A001 4C0C 00F0          A1 L = A1                % CLEAR PAR                              02269000 D
064E  4809 AC5C 9D3C 00F0          A1 OR B = A1, BMI        % RESTORE B AND SET PAR TO 1024          02270000 D
064F  482D 00C0 C000 00F0          JUMP                                                              02271000 D
                                                           %                                        02272000 D
                                                           %                                        02273000 D
                               PAGING:                     % THIS ROUTINE WILL CALCULATE AN ACTUAL   02274000 D
                                                           % ADDRESS WHEN PASSED A PAGE NUMBER       02275000 D
                                                           % REFERENCE AND AN OFFSET IN THE BR2 REG. 02276000 D
                                                           % A1, A2, A3, AMPCR MUST BE SAVED TO BE USED02277000 D
```

```
0650  4809 00C1 0000 00F0          L L = B                    % GENERATE 512 IN B          02278C00 D
0651  B000 0000 0000 0020          COMP 9 = SAR                                            02279000 D
0652  4809 2C46 0B00 00F0          LIT + B + 1 = B            % B NOW CONTAINS 768(ADDR OF PAGEREG) 0228CC00 D
0653  0FFC 0003 0C00 00E0          255 = LIT                                               02281C00 D
0654  4809 0CC0 0030 24F0          ASE                        % REFERENCE BR2               02282C00 D
0655  4809 0F42 8075 00F0          NOT BMAR R = CTR           % SAVE MAR(OFFSET) IN COUNTER 02283C00 D
0656  0000 00C0 0C00 001C          B = SAR                                                 02284C00 D
0657  4809 0F40 801C 00F0          BMAR R = MAR2              % PUT BR2(PAGE ADDR REG NO.) IN MAR2 02285C00 D
0658  0000 00C0 0C00 0020          16 = SAR                                                02286C00 D
0659  4809 C0C0 0030 00F0          A2 = MIR                   % SAVE A2 IN MIR             02287C0C D
065A  4809 0F40 200C 00F0          BMAR  = A2                 % USE A2 AS X-SELECT HOLDER FOR BMAR 02288C00 D
065B  4809 CC40 001C 00FC          A2 + B = MAR2              % MAR2 CONTAINS THE BASE ADDRESS OF THE 02289000 D
                                                              % DESIRED PAGE ADDRESS REGISTER C229CC00 D
065C  A809 C000 0000 0CF0          MR2, IF RDC                % READ CONTENTS OF PAGE ADDR. REG. 0229 1000 D
065D  AC08 0000 0C00 00F0          WHEN RDC THEN BEX          % READ COMPLETE, STORE IN B  02292C00 D
065E  4809 0C40 2000 0CF0          B = A2                     % SAVE PAGE ADDRESS REG CONTENTS IN A2 02293C00 D
065F  4809 18C0 8B00 00F0          B100 R = B                 % SET PAGE MODIFICATION BIT (BIT 15) 02294C0C D
0660  0C0C 00C0 0000 0020          16 = SAR                                                02295C00 D
0661  2BCB C000 C00C 00FC          IF LC1 THEN SET LC1, STEP ELSE SKIP  % STORE TO MEMORY, SET 02296000 D
                                                              % PAGE REFERENCED BIT IN PAGE ADDR REG 0229700C D
0662  4809 CC5C 200C 00F0          A2 OR B = A2               % FORM PAGE ADDRESS REG PLUS MOD. BIT 02298C00 D
0663  4805 C000 0D30 00F0          A2 = MIR,BMI               % SAVE RETURN IN B            02299C00 D
                                                              % MODIFIED PAGE ADDR. REG. IN MIR 02300C00 D
0664  3809 0000 0070 1CFC          MW2, IF SAI                % WRITE OUT MODIFIED PAGE REG  0230 1C0C D
0665  9C08 00C3 0000 00F0          WHEN SAI THEN 0                                         02302C00 D
0666  4809 C156 2C0C 00F0          A2 AND LIT = A2            % ISOLATE PAGE NUMBER         02303C00 D
0667  0FFC 0003 0030 00E0          255 = LIT                  % MASK OFF LOWER 8 BITS       02304C00 D
0668  4809 C0C3 0030 C0F0          A2 = MIR                   % RESTORE A2                  02305C00 D
0669  4809 0C40 2D0C 00F0          B = A2,BMI                 % B HAS THE PAGE REG. CONTENTS 02306C00 D
066A  4809 6C5D 8C10 00F0          CTR OR B C = BR2           % BR2 CONTAINS DIRECT ADDRESS (PAGING 02307C0C D
066B  000C 00C0 0C00 0020          16 = SAR                   % COMPLETED)                  02308C00 D
066C  2BCB C000 CC00 00FC          IF LC1 THEN SET LC1, STEP  ELSE SKIP  % TO CALLING ROUTINE 02309C0C D
066D  5260 00C0 003C 004C          PAGEOUT - 1 = MPCR         % RETURN TO EMULOUT           0230AC00 D
066E  6130 C000 0000 C040          PAGEIN - 1 = MPCR          % RETURN TO EMULIN            02311C00 D
                                                              %                             02312C00 D
                                                              %                             02313C0C D
                                                              %                             02314C00 D
                                 % * * * * * * * * * * * * * OPCODE IMPLEMENTATION * * * * * * * * * * * * * 02315C00 D
                                                              %                             02316C00 D
                                 OPCODE:                      % THIS ROUTINE WILL FETCH AN INSTRUCTION 02317C00 D
                                                              % ANALYZE THE OPCODE(UPPER  6 BITS)  AND02318C00 D
                                                              % SELECT THE CORRESPONDING ROUTINE FOR  02319C00 D
                                                              % THE SELECTED OPERATION.      0232CC00 D
066F  4809 A0C0 0000 C0F0          A1                         % A1 TO THE ADDER              02321C00 D
0670  4808 0003 C000 00F0          IF MST THEN STEP ELSE SKIP  % EXAMINE REMOTE EXECUTE BIT  02322C00 D
0671  5410 00C3 009C C06C          RESTPSW - 1 = CPCR                                      02323C00 D
0672  633C 0CC0 0030 006C          IFETCH - 1 = CPCR          % GET THE NEXT INSTRUCTION    02324C00 D
0673  2809 CCC0 0030 00F0          IF LC1                     % CLEAR CONDITION BITS         02325C0C D
0674  3809 0003 0C3C 00F0          IF LC2                                                  02326C00 D
0675  4809 0C43 1000 00F0          B = A3                     % SAVE INSTRUCTION INTO A3     02327C0C D
0676  4809 0C40 AC00 00FC          B R = A2                   % PLACE THE 6 OPCODE BITS IN LOW ORDER 6 02328C00 D
0677  200C 00C3 0030 0010          10 = SAR                   % BITS OF A2                  02329C00 D
0678  4809 C643 009C C0FC          A2 + AMPCR = AMPCR         % CREATE ADDRESS OF OPERATION ROUTINE EY 0233CC0C D
0679  0C20 00C0 0C3C C0C0          OPTABLE - 1 = AMPCR        % BASE ADDRESS OF OPTABLE      02331C00 D
067A  4809 0000 003C 00F0          STEP                       % ADDING OPCODE VALUE TO OPTABLE BASE 02332C00 D
067B  4824 00C0 0000 00F0          EXEC                       % JUMP TO AMPCR + 1            02333C0C D
                                                              %                             02334C00 D
                                                              %                             02335000 D
                                 OPTABLE:                     % THE FOLLOWING 64 INSTRUCTIONS CORRES- 02336C00 D
                                                              % POND TO SUPROUTINE CALLS TO THE      02337C00 D
```

```
                                                % A1 CONTAINS THE FSW                                    02339C00 D
                                                % A2 SERVES AS A TEMPORARY REGISTER                      02340C00 D
                                                % FOR RETURNS, ETC.                                     02341C00 D
                                                % P IS THE WORKING REGISTER                              02342C00 D
                                                % A3 CONTAINS THE INSTRUCTION                            02343C00 D
067C  0C30 0003 0000 0040      OPCODE00 - 1 = MFCR                                                      02344C00 D
067D  0C40 0003 0000 0040      OPCODE01 - 1 = MPCR                                                      02345C00 D
067E  0C50 0003 0000 0040      OPCODE02 - 1 = MFCR                                                      02346C00 D
067F  0C60 0003 0000 0040      OPCODE03 - 1 = MPCR                                                      02347C00 D
0680  0C70 0003 0000 004C      OPCODE04 - 1 = MFCR                                                      02348000 D
0681  0C80 0003 0000 0040      OPCODE05 - 1 = MPCR                                                      02349C00 D
0682  0C90 0003 0000 004C      OPCODE06 - 1 = MPCR                                                      02350C00 D
0683  0CA0 0003 0000 004C      OPCODE07 - 1 = MFCR                                                      02351C00 D
0684  0CB0 0003 0000 0040      OPCODE10 - 1 = MPCR                                                      02352C00 D
0685  0CC0 0003 0000 0040      OPCODE11 - 1 = MPCR                                                      02353C00 D
0686  0CD0 0003 0000 004C      OPCODE12 - 1 = MPCR                                                      02354C00 D
0687  0CE0 0003 0000 0040      OPCODE13 - 1 = MFCR                                                      02355C00 D
0688  0CF0 0003 0000 0040      OPCODE14 - 1 = MPCR                                                      02356C00 D
0689  0D00 0003 0000 0040      OPCODE15 - 1 = MPCR                                                      02357C00 D
068A  0D10 0003 0000 004C      OPCODE16 - 1 = MFCR                                                      02358C00 D
068B  0D20 0003 0000 0040      OPCODE17 - 1 = MPCR                                                      02359C00 D
068C  0D30 0003 0000 0040      OPCODE20 - 1 = MPCR                                                      02360C00 D
068D  0D40 0003 0000 0040      OPCODE21 - 1 = MPCR                                                      02361C00 D
068E  0D50 0003 0000 0040      OPCODE22 - 1 = MPCR                                                      02362C00 D
068F  0D60 0003 0000 004C      OPCODE23 - 1 = MPCR                                                      02363C00 D
0690  0D70 0003 0000 0040      OPCODE24 - 1 = MFCR                                                      02364C00 D
0691  0D80 0003 0000 0040      OPCODE25 - 1 = MFCR                                                      02365C00 D
0692  0D90 0003 0000 0040      OPCODE26 - 1 = MPCR                                                      02366C00 D
0693  0DA0 0003 0000 004C      OPCODE27 - 1 = MPCR                                                      02367C00 D
0694  0DB0 0003 0000 004C      OPCODE30 - 1 = MFCR                                                      02368000 D
0695  0DC0 0003 0000 0040      OPCODE31 - 1 = MFCR                                                      02369C00 D
0696  0DD0 0003 0000 0040      OPCODE32 - 1 = MPCR                                                      02370C00 D
0697  0DE0 0003 0000 0040      OPCODE33 - 1 = MFCR                                                      02371C00 D
0698  0DF0 0003 0000 0040      OPCODE34 - 1 = MPCR                                                      02372C00 D
0699  0E00 0003 0000 0040      OPCODE35 - 1 = MPCR                                                      02373C00 D
069A  0E10 0003 0000 0040      OPCODE36 - 1 = MFCR                                                      02374C00 D
069B  0E20 0003 0000 0040      OPCODE37 - 1 = MPCR                                                      02375C00 D
069C  0E30 0003 0000 0040      OPCODE40 - 1 = MPCR                                                      02376C00 D
069D  0E40 0003 0000 0040      OPCODE41 - 1 = MFCR                                                      02377C00 D
069E  0E50 0003 0000 0040      OPCODE42 - 1 = MPCR                                                      02378C00 D
069F  0E60 0003 0000 0040      OPCODE43 - 1 = MPCR                                                      02379C00 D
06A0  0E70 0003 0000 0040      OPCODE44 - 1 = MPCR                                                      02380C00 D
06A1  0E80 0003 0000 004C      OPCODE45 - 1 = MPCR                                                      02381C00 D
06A2  0E90 0003 0000 0040      OPCODE46 - 1 = MPCR                                                      02382C00 D
06A3  0EA0 0003 0000 0040      OPCODE47 - 1 = MPCR                                                      02383C00 D
06A4  0EB0 0003 0000 0040      OPCODE50 - 1 = MFCR                                                      02384C00 D
06A5  0EC0 0003 0000 0040      OPCODE51 - 1 = MFCR                                                      02385C00 D
06A6  0ED0 0003 0000 004C      OPCODE52 - 1 = MPCR                                                      02386C00 D
06A7  0EE0 0003 0000 0040      OPCODE53 - 1 = MPCR                                                      02387C00 D
06A8  0EF0 0003 0000 0040      OPCODE54 - 1 = MFCR                                                      02388C00 D
06A9  0F00 0C00 0C30 0040      OPCODE55 - 1 = MFCR                                                      02389000 D
06AA  0F10 0003 0000 0040      OPCODE56 - 1 = MPCR                                                      02390C00 D
06AB  0F20 0003 0000 0040      OPCODE57 - 1 = MPCR                                                      02391C00 D
06AC  0F30 0003 0000 004C      OPC9DE6C - 1 = MFCR                                                      02392C00 D
06AD  0F40 0003 0000 0040      OPCODE61 - 1 = MFCR                                                      02393000 D
06AE  0F50 0003 0000 0040      OPCODE62 - 1 = MFCR                                                      02394C00 D
06AF  0F60 0003 0C30 004C      OPCODE63 - 1 = MPCR                                                      02395C00 D
06B0  0F70 0003 0000 0040      OPCODE64 - 1 = MPCR                                                      02396C00 D
06B1  0F80 0003 0000 0040      OPCODE65 - 1 = MPCR                                                      02397C00 D
```

```
06B2  0F9C 0DCD 0090 0040          OPCODE66 - 1 = MPCR                                  02398C00 D
06B3  0FAC 0DC9 0090 0040          OPCODE67 - 1 = MPCR                                  02399CC0 D
06B4  0FBC 0CCD 0090 0040          OPCODE70 - 1 = MPCR                                  02400C00 D
06B5  0FCC 0CC9 009C 0040          OPCODE71 - 1 = MPCR                                  02401C90 D
06B6  0FDC 0DC9 009C 0040          OPCODE72 - 1 = MPCR                                  02402C00 D
06B7  0FEC 0DCD 0090 004C          OPCODE73 - 1 = MPCR                                  02403000 D
06B8  0FFC 0DC9 0C90 C04C          OPCODE74 - 1 = MPCR                                  02404C0C D
06B9  1000 0DCD 009C 004C          OPCODE75 - 1 = MPCR                                  02405C00 D
06BA  101C 0DC9 0C90 004C          OPCODE76 - 1 = MPCR                                  02406000 D
06BB  102C 0CC9 009C 0040          OPCODE77 - 1 = MPCR                                  02407000 D
                                                                 %                      02408C00 D
                                                                 %                      02409C0C D
                                   OPCODE00:                     % THIS ROUTINE ANALYZES OC OPCODE    02410000 D
06BC  5F90 0DCD 0C9C C06C                XFCODE - 1 = CPCR       % "F" INTO A2                        02411C00 D
06BD  4809 C64D 0C9C C0F0                A2 + AMPCR = AMPCR      % JUMP TO DESIRED SUBFUNCTION        02412000 D
06BE  1030 0CCD 0090 00C0                OPODCF - 1 = AMPCR      % SET UP BASE ADDR OF TYPECO INST.   02413C00 D
06BF  4809 0DC9 009C C0F0                STEP                                                 02414C00 D
06C0  4824 0DCD 009C C0F0                EXEC                                                 02415C00 D
                                                                 %                      02416C00 D
                                                                 %                      02417000 D
06C1  4E5C 0DCD 009C 0040          OP00F:  NOTIMP - 1 = MPCR                                  02418C00 D
06C2  300C 0DCD 0090 0C40                  FAULT - 1 = MPCR                                   02419C00 D
06C3  300C 0DC9 009C 0040                  FAULT - 1 = MPCR                                   02420C90 D
06C4  104C 0DCD C090 0040                  OP003 - 1 = MPCR                                   02421C00 D
                                                                 %                      02422CCC D
                                                                 %                      02423000 D
                                   OP003:                        % RX TYPE, BYTE                      02424C00 D
                                                                 % THIS ROUTINE STORES THE DESIGNATED BYTE02425C00 D
                                                                 % OF Y INTO BITS C-7 OF R(A), AND CLEARS 02426C00 D
                                                                 % THE REMAINING BITS OF R(A)          02427000 D
06C5  4849 00CD 00FC 00F0                SET LC2                 % SET FLAG FOR BYTE INST.            02428000 D
06C6  5700 00C9 0090 0060                RXMFIELD - 1 = CPCR     % B = Y                              02429C00 D
06C7  4809 0C41 0010 00F0                B L = BR2                                            02430C00 D
06C8  0000 0000 CC9C C030                COMP 8 = SAR            % ADDR. OF Y IN BR2                  02431000 D
06C9  60E0 C00D 0090 0060                EMULIN - 1 = CPCR       % B = (Y)                            02432000 D
06CA  300B 00C0 000F 00FC                IF NOT LC2 THEN STEP ELSE SKIP  % MS BYTE OF (Y)             02433C0C D
06CB  4809 0C4D 8B90 00F0                B R = B                 % OR LS BYTE OF (Y)                  02434C00 D
06CC  0FFC 0003 0090 C090                B = SAR; 255 = LIT                                   02435000 D
06CD  4809 2C56 0B90 00F0                LIT AND B = B;MIR       % NEW CONTENTS OR R(A)               02436C0C D
06CE  200C 0003 0000 006C                SETCCA - 1= CPCR        % ARTHMETIC CC SETTING (ENTER WITH 9) 02437CC0 D
06CF  4809 E000 8B9C C0F0                A3 R = B                % ISOLATE R(A) IN LS 4 BITS           02438CC0 D
06D0  80FC 0009 0C90 C080                4 = SAR; 15 = LIT                                    02439C0C D
06D1  48C9 2C56 CB3C 00FC                LIT AND B = B                                        02440CDC D
06D2  51CC 00C0 C000 0060                REGSTACK - 1 = CPCR                                  02441000 D
06D3  2F5C 00C3 0090 0060                EOUTPUT - 1 = CPCR                                   02442C00 D
06D4  66EC 0DCD 0000 0040                OPCODE - 1 = MPCR                                    02443C00 D
                                                                 %                      02444C0C D
                                                                 %                      02445CC0 D
                                   OPCODE01:                     % THIS ROUTINE WILL ANALYZE 01 OPCODES.  02446C0C D
                                                                 % THE INSTRUCTION MUST BE PASSED VIA THE 02447000 D
                                                                 % B REGISTER.                        02448C0C D
06D5  5F90 0000 0090 006C                XFCODE - 1 = CPCR       % "F" INTO A2                        02449C0C D
06D6  4809 C640 0040 00F0                A2 + AMPCR = AMPCR      % ADDR OF DESIRED SUBFUNCTION OF OPCODE  02450C00 D
06D7  105C 00C3 C00C 00CC                OP01F - 1 = AMPCR       % SET UP BASE ADDR OF TYPE 01 INSTRUCTION02451C0C D
06D8  4809 0000 0C90 00F0                STEP                    % 01 IS DETERMINED BY ADDING THE FUNCTION02452C00 D
06D9  4824 0000 009C 00F0                EXEC                    % FIELD CODE.                        02453C0C D
                                                                 %                      02454CC0 D
                                                                 %                      02455000 D
06DA  106C 00C0 0090 0040          OP01F:  OP010 - 1 = MPCR       % THIS ROUTINE JUMPS TO THE APPROPRIATE  02456C0C D
06DB  1070 00C0 0C90 C040                  OP011 - 1 = MPCR       % SUBFUNCTION FOR OPCODE 01          02457C0C D
```

```
06DC  1080 0000 0000 0040          OP012 - 1 = MPCR                                    02458C00 D
06E0  1090 0000 0C30 0040          OP013 - 1 = MPCR                                    02459C00 D
                                                                                       02460C00 D
                                                                                       02461C00 D
                                   OP010:                                              02462C00 D
                                                                                       02463C00 D
06DE  4809 2C56 0B0C 00FC          B AND LIT = B                                       02464C00 D
06DF  00F0 C000 0070 00EC          15 = LIT                                            02465C00 D
06E0  51CC C0C3 0C00 0060          REGSTACK - 1 = CPCR                                 02466C00 D
                                                                                       02467C00 D
06E1  2F30 0000 0000 0060          EINPUT - 1 = CPCR                                   02468C00 D
06E2  4809 0C40 0030 00F0          B = MIR                                             02469C00 D
06E3  200C 0000 0C3C C06C          SETCCA - 1= CPCR                                    02470000 D
06E4  4809 E0C0 9000 00F0          A3 R = A3                                           02471C0C D
06E5  80FC 00C3 0000 0080          4 = SAR, 15 = LIT                                   02472C00 D
06E6  4809 E156 0E00 00FC          A3 AND LIT = D                                      02473C00 D
06E7  51CC 0003 0070 006C          REGSTACK - 1 = CPCR                                 02474C00 D
                                                                                       02475CC0 D
06E8  2F50 0C00 000C 0060          EOUTPUT - 1 = CPCR                                  02476F00 D
06E9  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                   02477C00 D
                                                                                       02478C00 D
                                                                                       02479C00 D
                                   OP011:                                              02480C00 D
06EA  4809 2C56 0B30 00F0          B AND LIT = B                                       02481C00 D
06EB  00F0 0000 0070 00E0          15 = LIT                                            02482C0C D
06EC  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                                 02483C00 D
                                                                                       02484C00 D
06ED  2F30 0000 000C 0060          EINPUT - 1 = CPCR                                   02485CC0 D
06EE  4809 0C41 0C10 C0FC          B L = BR2                                           02486C00 D
06EF  000C 0000 0000 003C          COMP 0 = SAR                                        02487000 D
06F0  60E0 0000 C070 0060          EMULIN - 1 = CPCR                                   02488C00 D
06F1  4809 0C40 0030 00F0          B = MIR                                             02489C00 D
06F2  200C 0000 0000 0060          SETCCA - 1= CPCR                                    02490C00 D
06F3  4809 E0C0 B80C C0FC          A3 R = B                                            02491C00 D
06F4  80F0 00C3 000C 0080          4 = SAR, 15 = LIT                                   02492C00 D
06F5  4809 2C56 0B00 00FC          B AND LIT = B                                       02493C00 D
06F6  51CC 0000 0000 C060          REGSTACK - 1 = CPCR                                 02494C00 D
                                                                                       02495C00 D
06F7  2F50 C000 000C 0060          EOUTPUT - 1 = CPCR                                  02496C0C D
06F8  66E0 C000 0C3C C04C          OPCODE - 1 = MPCR                                   02497C00 D
                                                                                       02498C00 D
                                                                                       02499C00 D
                                   OPC12:                                              02500C00 D
                                                                                       02501C00 D
                                                                                       02502C00 D
06F9  6330 00C0 0030 006C          IFETCH - 1 = CPCR                                   02503000 D
06FA  4809 E0C1 103C 00FC          A3 L = A3                                           02504C00 D
06FB  00F0 0FC0 0030 00A0          COMP 16 = SAR, 15 = LIT                             02505C0C D
06FC  4809 EC5C 1C0C 00F0          A3 OR B = A3                                        02506C00 D
06FD  4809 0000 0C30 C0F0          C = MIR,BR2                                         02507C00 D
06FE  4809 E0C0 A6D0 C0F0          A3 R = A2                                           02508C00 D
06FF  4809 C156 001C C0FC          A2 AND LIT = MAR2                                   02509C00 D
0700  4809 0F52 000C 00F0          BMAR NEQ 0                                          02510C00 D
0701  600B 0000 0C00 C0F0          IF TRUE THEN STEP ELSE SKIP                         02511C00 D
0702  567C 0003 0000 006C          RM - 1 = CPCR                                       02512C00 D
0703  4809 E0C1 2C30 C0F0          A3 L = A2                                           02513C00 D
0704  000C 00C0 0000 C020          COMP 16 = SAR                                       02514C00 D
0705  4809 C000 AD00 C0F0          A2 R = A2, BMI                                      02515C00 D
0706  4809 CC40 0B3C 00FC          A2 + B= G,MIR                                       02516C00 D
0707  200C 0000 0000 0060          SETCCA - 1= CPCR                                    02517C00 D
```

```
0708  4809 E000 8830 00F0          A3 R = B                    % PUT :A: FIELD IN LS 4 BITS OF B        02518C00 D
0709  80F0 0003 0030 C0A0          20 = SAR; 15 = LIT                                                  02519C00 D
070A  4809 2C56 0B30 00F0          LIT AND B = B               % PUT ADDR OF R(A) INTO B                02520C00 D
070B  51C0 00C0 0C90 0060          REGSTACK - 1 = CPCR         % SELECT REGISTER STACK TO BE USED       02521C00 D
                                                               % ADDR OF GEN REG STACK RETURNED IN MAR2 02522C00 D
070C  2F5C 0000 0090 0060          EOUTPUT - 1 = CPCR          % (R(M)) + (Y) INTO R(A)                 02523C00 D
070D  66E0 00C0 0070 0040          OPCODE - 1 = MPCR           % GET NEXT INSTRUCTION                   02524C00 D
                                                               %                                        02525C00 D
                                                               %                                        02526C00 D
                              OP013:                           % RX TYPE INSTRUCTION                    02527C00 D
                                                               % (Y) INTO R(A)                          02528C00 D
070E  5700 C000 0C00 C060          RXMFIELD - 1 = CPCR         % ANALYZE "M" FIELD CONTENTS             02529C00 D
070F  4809 0C41 0010 00F0          B L = BR2                   % ISOLATE Y INTO BR2                     02530C00 D
0710  0000 C0C0 0090 0030          COMP B = SAR                                                        02531C00 D
0711  60E0 0000 0000 0060          EMULIN - 1 = CPCR           % (Y) INTO B                            02532C00 D
0712  4809 0C40 009C 00F0          B = MIR                     % STORE (Y) INTO MIR                     02533C00 D
0713  2000 00C0 0000 0060          SETCCA - 1= CPCR            % ARITHMETIC CC SETTING (ENTER WITH B)   02534C00 D
0714  4809 E000 8800 00FC          A3 R = B                                                            02535C00 D
0715  80FC 0003 0C00 C080          4 = SAR; 15 = LIT                                                   02536C00 D
0716  4809 2C56 0B3C 00F0          LIT AND B = B               % ISOLATE :A: FIELD                      02537C00 D
0717  51C0 00C0 0000 0060          REGSTACK - 1 = CPCR         % SELECT REGISTER STACK TO BE USED       02538C00 D
                                                               % ADDR OF GEN REG STACK RETURNED IN MAR2 02539C0C D
0718  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR                                                  02540C00 D
0719  66E0 00C0 0000 C040          OPCODE - 1 = MPCR                                                   02541C00 D
                                                               %                                        02542C00 D
                                                               %                                        02543C00 D
                              OPCODE02:                        % THIS ROUTINE  DECODES THE 02 OPCODE    02544000 D
071A  2809 00C0 0030 00F0          IF LC1                                                              02545C00 D
071B  3809 00C0 C030 00F0          IF LC2                      % CLEARS LOCAL CONC. BITS                02546000 D
071C  5F90 00C0 0000 0060          XFCODE - 1 = CPCR                                                   02547C00 D
071D  4809 C64D 0090 00F0          A2 + AMPCR = AMPCR                                                  02548C00 D
071E  10AC 00C0 0000 00C0          OP02F - 1 = AMPCR                                                   02549C00 D
071F  4809 0000 0000 00F0          STEP                                                                02550C00 D
0720  4824 0003 0000 00F0          EXEC                                                                02551C00 D
                                                               %                                        02552C00 D
0721  1080 0000 0000 0040    OP02F:  OP020 - 1 = MPCR                                                  02553C00 D
0722  10C0 0000 0000 0040          OP021 - 1 = MPCR                                                    02554C00 D
0723  300C 00C0 0030 C040          FAULT - 1 = MPCR                                                    02555C00 D
0724  1000 00C0 0000 C040          OP023 - 1 = MPCR                                                    02556C00 D
                                                               %                                        02557C00 D
                              OP020:                           % UNARY FUNCTION DESCRIPTOR             02558C00 D
                                                               % :M: FIELD  SELECTS UNARY INST.         02559C00 D
0725  4009 CC40 8800 00F0          B R = B                     % ISOLATE :A: FIELD                      02560C00 D
0726  90F0 00C3 C090 0080          4 = SAR; 15 = LIT                                                   02561C00 D
0727  4809 2C56 0B3C 00F0          LIT AND B = B                                                       02562C00 D
0728  51C0 00C3 0000 0060          REGSTACK - 1 = CPCR                                                 02563C00 D
0729  2F3C 0003 0C90 C060          EINPUT - 1 = CPCR           % B = (R(A)); MAR2 = RA                 02564C0C D
072A  4809 E155 2C30 C0F0          A3 AND LIT = A2                                                     02565C00 D
072B  00FC 0000 0C00 C0EC          15 = LIT                    % ISOLATE :M: FIELD                      02566C00 D
072C  4809 C64D 0090 00FF          A2 + AMPCR = AMPCR                                                  02567C00 D
072D  10EC 00C0 0C00 C0C0          OP02CM - 1 = AMPCR                                                  02568C0C D
072E  4809 C155 0030 00F0          A2 GEQ LIT                  % TEST FOR NOT IMPLEMENTED BOUNDRY       02569C00 D
072F  00C0 00C0 0030 C0E0          12 = LIT                                                            02570C00 D
0730  780B 0000 0000 00F0    IF TRUE THEN STEP ELSE SKIP                                               02571C00 D
0731  3000 00C0 0030 0040          FAULT - 1 = MPCR                                                    02572C00 D
0732  4824 0003 0000 00F0          EXEC                                                                02573C00 D
                                                               %                                        02574C0C D
0733  10F0 00C0 0030 0040    OP020M:  OP02000 - 1 = MPCR                                               02575C00 D
0734  1100 0003 0000 0040          OP02001 - 1 = MPCR                                                  02576C00 D
0735  1110 0003 0000 0040          OP02002 - 1 = MPCR                                                  02577C00 D
```

```
0736  3000 0000 0000 0040          FAULT - 1 = MPCR                                    02578000 D
0737  1120 0000 0000 0040          OP02004 - 1 = MPCR                                  02579000 D
0738  1130 0000 0000 0040          OP02005 - 1 = MPCR                                  02580000 D
0739  1140 0000 0000 0040          OP02006 - 1 = MPCR                                  02581000 D
073A  3000 0000 0000 0040          FAULT - 1 = MPCR                                    02582000 D
073B  1150 0000 0000 004C          OP02010 - 1 = MPCR                                  02583000 D
073C  1160 0000 0000 004C          OP02011 - 1 = MPCR                                  02584000 D
073D  1170 0000 0000 0040          OP02012 - 1 = MPCR                                  02585000 D
073E  1180 0000 0000 0040          OP02013 - 1 = MPCR                                  02586000 D
                                                 %                                     02587000 D
                                   OP02000:      % TYPE RR; MAKE POSITIVE              02588000 D
                                                 % IF (R(A)) < 0, (R(A)): -> RA        02589000 D
                                                 % IF (R(A)) => 0, (R(A)) UNCHANGED    02590000 D
073F  4809 0041 2800 00F0          B L = A2,B                                          02591000 D
0740  0000 0000 0000 0020          COMP 16 = SAR           % SHIFT (R(A)) TO UHW OF A2,B 02592000 D
0741  4809 0812 0000 00F0          A2 EQL B100             % TEST FOR MAX NEG. NUMBER   02593000 D
0742  6BC9 00C0 0000 C0F0          IF TRUE THEN SET LC1                                02594000 D
0743  5C20 0000 0000 0060          OVBIT - 1 = CPCR        % SET OR CLEAR OV BIT       02595000 D
0744  4809 0003 0000 00F0          STEP                                                02596000 D
0745  0000 0003 0000 0020          16 = SAR                                            02597000 D
0746  4809 C003 C000 C0FC          A2                      % A2 TO THE ADDER           02598000 D
0747  400B 0900 0000 00F0          IF NOT MST THEN STEP ELSE SKIP                      02599000 D
0748  66E0 0003 0C00 0040          OPCODE - 1 = MPCR                                   02600000 D
0749  4849 0C5E 8B90 C0F0          -B R = B, MIR; SET LC2                              02601000 D
074A  7BC9 00C3 0020 C0F0          IF AOV THEN SET LC1     % CHECK FOR CARRY           02602000 D
074B  1E70 00C0 0C30 C06C          CARRY - 1 = CPCR                                    02603000 D
074C  38CB 0CC3 0020 00FC          IF LC2 THEN STEP ELSE SKIP  % LC2 INDICATES WHETHER RA 02604000 D
074D  2F5C 0003 0000 C060          EOUTPUT - 1 = CPCR      % MUST BE UPDATED           02605000 D
074E  20D0 00C0 0030 0060          SETCCA - 1 = CPCR       % B CONTAINS (RA)           02606000 D
074F  66E0 00C3 0000 0040          OPCODE - 1 = MPCR                                   02607000 D
                                                 %                                     02608000 D
                                   OP02001:      % TYPE RR; MAKE NEGATIVE             02609000 D
                                                 % IF (R(A)) > 0, (R(A)): -> RA        02610000 D
                                                 % IF (R(A)) <= 0, (R(A)) UNCHANGE     02611000 D
0750  4809 0C41 2B90 0CF0          B L = A2,B              % (R(A)) INTO UHW OF A2,B    02612000 D
0751  0000 0003 0030 C020          COMP 16 = SAR                                       02613000 D
0752  4809 C000 0030 C0F0          A2                      % A2 TO THE ADDER           02614000 D
0753  4419 C012 0C00 00F0          IF NOT MST THEN A2 EQL 0; SKIP                      02615000 D
0754  56E0 0000 0CC0 C040          OPCODE - 1 = MPCR                                   02616000 D
0755  580B 0C00 0030 00F0          IF TRUE THEN STEP ELSE SKIP                         02617000 D
0756  56E0 0CC0 000C 004C          OPCODE - 1 = MPCR                                   02618000 D
0757  4849 0C5E 8B90 0CFC          -B R = B, MIR; SET LC2                              02619000 D
0758  7BC9 0CC0 000C 00F0          IF AOV THEN SET LC1     % CHECK FOR CARRY           02620000 D
0759  1E70 00C0 0000 C060          CARRY - 1 = CPCR                                    02621000 D
075A  380B 00C0 0C00 C0FC          IF LC2 THEN STEP ELSE SKIP                          02622000 D
075B  2F50 00C0 009C 006C          EOUTPUT - 1 = CPCR      %(R(A)): -> RA ONLY IF LC2 IS SET 02623000 D
075C  20D0 0000 0030 0060          SETCCA - 1 = CPCR       % B = (R(A)) OR (R(A)):     02624000 D
075D  56E0 C0C0 0C00 C040          OPCODE - 1 = MPCR                                   02625000 D
                                                 %                                     02626000 D
                                   OP02002:      % ROUND TYPE RR                       02627000 D
                                                 % IF (R(A)) >= 0 (R(A))+(R(A+1))15 -> RA 02628000 D
                                                 % IF (R(A)) < 0 (R(A))-(R(A+1))15 -> RA 02629000 D
                                                 % RA MUST BE EVEN                     02630000 D
075E  4809 0C41 209C 00F0          B L = A2                % B = (R(A)); MAR2 = RA      02631000 D
075F  0010 0CC3 000C 00AC          COMP 16 = SAR; 1 = LIT                              02632000 D
0760  4809 2F5C 0B9C 00F0          LIT OR BMAR = B         % R(A+1)                    02633000 D
0761  51C0 0000 0090 C060          REGSTACK - 1 = CPCR     % R(A+1) INTO MAR2          02634000 D
0762  2F3C 0000 C000 006C          EINPUT - 1 = CPCR       % B = (R(A+1))              02635000 D
0763  4809 C0C0 0030 00F0          A2                                                  02636000 D
0764  4819 0003 0C90 C0FC          IF MST THEN SKIP                                    02637000 D
```

```
0765  119C 00C0 003C 0040          RPOS - 1 = MPCR                                  02638000 D
                                                                                    02639000 D
0766  4809 0C42 0B0C 00F0   RNEG:  NOT B = B                 % COMPLEMENT B (1:S)    02640000 D
0767  4809 0C40 8B0C 00F0          B R = B                  % ISOLATE BIT 15 IN B   02641000 D
0768  B00C 00C0 003C 0010          15 = SAR                                         02642000 D
0769  4809 0C41 0B3C 00F0          B L = B                  % CLEAN OUT B REGISTER  02643000 D
076A  100C C00C 003C 0000          COMP 31 = SAR                                    02644000 D
076B  4809 0C40 8D0C C0F0          B R = B                  % PLACE BIT 15 IN BIT 16 02645000 D
076C  B000 00C0 003C C010          15 = SAR                                         02646000 D
076D  4849 CC5E 0030 00F0          A2 - B = MIR; SET LC2                            02647000 D
076E  7BC9 0000 0000 C0F0          IF AOV THEN SET LC1      % CHECK FOR CARRY       02648000 D
076F  1E70 00C0 003C 0060          CARRY - 1 = CPCR                                 02649000 D
0770  4F10 00C0 003C C060          CHECKOV - 1 = CPCR       % OVERFLOW CHECK        02650000 D
0771  11AC 00C0 0C7C 004C          END02002 - 1 = MPCR                              02651000 D
                                                                                    02652000 D
0772  48C9 0C41 0D0C C0F0   RPOS:  D L = B                  % SHIFT BIT POS 15 OF RA+1 INTO BIT 16  02653000 D
0773  B000 00C0 003C 0030          COMP 1 = SAR                                     02654000 D
0774  4809 CC40 CC3C 00F0          A2 + B = MIR                                     02655000 D
0775  7DC9 C0C0 003C 00FC          IF AOV THEN SET LC1      % CHECK FOR CARRY       02656000 D
0776  1E70 C0C0 003C C060          CARRY - 1 = CPCR                                 02657000 D
0777  4F1C 00C0 003C 006C          CHECKOV - 1 = CPCR       % CHECK FOR OVERFLOW    02658000 D
0778  4809 00C0 003C 24F0   END02002:ASE                                           02659000 D
0779  4809 0F40 203C 00F0          BMAR = A2                % SET UP R(A) INTO MAR2  02660000 D
077A  4809 C0CE 001C 00F0          A2 AND B110 = MAR2, BMI % FORM RA                02661000 D
077B  4809 CC4D 8D3C 00F0          B R = B,MIR              % RESULT  (R(A)) STORE INTO LHW OF B,MIR 02662000 D
077C  0C00 0000 0C3C 0020          16 = SAR                                         02663000 D
077D  2C0C 00C0 003C C060          SETCCA - 1 = CPCR                                02664000 D
077E  2F50 00C0 003C 0060          EOUTPUT - 1 = CPCR                               02665000 D
077F  66EC 00C0 0C3C 0040          OPCODE - 1 = MPCR                                02666000 D
                                                                                    02667000 D
                              OP02004:                   % 2;S COMPLEMENT; TYPE RR  02668000 D
0780  4809 0C41 2B3C 00F0          B L = A2,B               % A2,B HAVE (R(A)) IN UHW; MAR2 = RA  02669000 D
0781  000C 00C0 0C3C 0020          COMP 16 = SAR                                    02670000 D
0782  4809 D812 0C3C C0F0          A2 EOL B100                                      02671000 D
0783  6DC9 00C0 003C C0F0          IF TRUE THEN SET LC1     % CHECK FOR MAX NEG. NUM.  02672000 D
0784  5G2C 00C0 003C C06C          OVBIT - 1 = CPCR         % SET/CLEAR OVBIT       02673000 D
0785  000C 00C3 0C3C 002C          16 = SAR                                         02674000 D
0786  48C9 0C5E 8B3C 00F0          - B R = B,MIR            % TWO;S COMPLEMENT (R(A))  02675000 D
0787  7BC9 00C0 0000 00F0          IF AOV THEN SET LC1      % CHECK FOR CARRY        02676000 D
0788  1E70 00C3 0C3C 0060          CARRY - 1 = CPCR                                 02677000 D
0789  2D0C C0C3 0C3C 0060          SETCCA - 1 = CPCR        % R = (R(A))            02678000 D
078A  2F50 00C3 0C3C C06C          EOUTPUT - 1 = CPCR                               02679000 D
078B  66EC 00C0 003C 004C          OPCODE - 1 = MPCR                                02680000 D
                                                                                    02681000 D
                              OP02005:                   % TWO;S COMPLEMENT DOUBLE; TYPE RR  02682000 D
                                                         % (RA,RA+1); -> RA,RA+1    02683000 D
078C  4809 0C00 0C3C 24F0          ASE                      % REF BR2               02684000 D
078D  4809 0F41 0C2C 00F0          BMAR L = BR1             % TEMP STORAGE OF R(A)   02685000 D
078E  000C 00C0 003C 003C          COMP 8 = SAR                                     02686000 D
078F  4809 0C41 203C C0F0          B L = A2                 % B = (R(A)); MAR2 = RA  02687000 D
0790  001C 0C00 003C 00A0          COMP 16 = SAR ; 1 = LIT                          02688000 D
0791  48C9 2F5C 001C 00F0          LIT OR BMAR = MAP2       % R(A+1)                02689000 D
0792  2F30 0000 0C3C C060          FINPUT - 1 = CPCR        % B = (R(A+1)) MAR2 = RA + 1  02690000 D
0793  4809 CC5C 2B3C 00F0          A2 OR B = A2, B          % A2 = (RA,RA+1)         02691000 D
0794  4809 D812 0C3C C0F0          A2 EOL B100                                      02692000 D
0795  6BC9 0CC0 003C 00FC          IF TRUE THEN SET LC1     % TEST FOR MAX NEG NUM  02693000 D
0796  5020 00C3 003C C06C          OVBIT - 1 = CPCR                                 02694000 D
0797  4809 0C5E 0B3C 00F0          - B = B                  % TWO;S COMPLEMENT OF (R(A),R(A+1))  02695000 D
0798  7BC9 00C3 0C3C 00F0          IF AOV THEN SET LC1      % CHECK FOR CARRY       02696000 D
0799  1E70 00C3 003C C060          CARRY - 1 = CPCR                                 02697000 D
```

```
079A  4809 0C41 100C 00F0          B L = A3                 % ISOLATE (R(A+1)) INTO MIR                 02698000 0
079B  0000 0000 0030 0020          COMP 16 = SAR                                                        02699000 0
079C  4809 E000 8030 C0F0          A3 R = MIR                                                           02700000 0
079D  2F5C 0000 0000 0060          FOUTPUT - 1 = CPCR                                                   02701000 0
079E  4809 0C40 8090 20F0          B R = MIR; ASR           % (R(A)) INTO MIR                           02702000 0
079F  49C9 0F43 801C 00F0          BMAR R = MAR2; SET LC1    % R(A); SET DOUBLE WORD ARITH COMPARE      02703000 0
07A0  000C 0000 0000 0010          B = SAR                                                              02704000 0
07A1  2F50 0000 000C C060          EOUTPUT - 1 = CPCR                                                   02705000 0
07A2  2000 0000 0000 0060          SETCCA - 1 = CPCR        % B CONTAINS (R(A),R(A+1))                  02706000 0
07A3  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                                    02707000 0
                                                            %                                           02708000 0
                          OP02006:                          % ONE:S COMPLEMENT; TYPE RR                 02709000 0
                                                            % (R(A)) BIT BY BIT -> RA                   02710000 0
                                                            % B = (R(A)); MAR2 = RA                     02711000 0
07A4  4809 0C43 0B30 00F0          NOT B L = B              % B = (R(A))                                02712000 0
07A5  0000 0000 0030 0020          COMP 16 = SAR                                                        02713000 0
07A6  4809 0C40 8B30 C0F0          B R = B, MIR                                                         02714000 0
07A7  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR                                                   02715000 0
07A8  2000 0000 0000 0060          SETCCA - 1 = CPCR                                                    02716000 0
07A9  66E0 0000 0030 0040          OPCODE - 1 = MPCR                                                    02717000 0
                                                            %                                           02718000 0
                          OP02010:                          % INCREMENT RA BY 1; TYPE RR                02719000 0
                                                            % (R(A)) + 1 -> RA; SET CC                  02720000 0
                                                            % B = (R(A)) ; MAR2 = RA                    02721000 0
07AA  4809 0C41 2030 C0F0          B L = A2                                                             02722000 0
07AB  0000 0000 0000 0020          COMP 16 = SAR            % (R(A)) IN UHW OF A2                       02723000 0
07AC  48C9 00C1 0B30 C0F0          1 L = B                  % INCREMENT  IN BIT 16                      02724000 0
07AD  4809 CC40 0030 00F0          A2 + B = MIR                                                         02725000 0
07AE  7BC9 0000 C030 00F0          IF ADV THEN SET LC1      % CHECK FOR CARRY                           02726000 0
07AF  1E70 00C0 0030 C060          CARRY - 1 = CPCR                                                     02727000 0
07B0  4F10 0000 0000 0060          CHECKOV - 1 = CPCR       % CHECK FOR OVERFLOW                        02728000 0
07B1  4809 0000 0000 00F0          BMI                                                                  02729000 0
07B2  4809 0C40 8B30 00F0          B R = B,MIR              % RESULT IN LHW OF B,MIR                    02730000 0
07B3  0000 C000 0030 C020          16 = SAR                                                             02731000 0
07B4  2000 C0C0 0000 006C          SETCCA - 1 = CPCR                                                    02732000 0
07B5  2F5C C0C0 0000 0060          EOUTPUT - 1 = CPCR                                                   02733000 0
07B6  66E0 0003 0030 C040          OPCODE - 1 = MPCR                                                    02734000 0
                                                            %                                           02735000 0
                          OP02011:                          % DECREMENT RA BY 1; TYPE RR                02736000 0
                                                            % (R(A)) - 1 -> RA                          02737000 0
                                                            % B = (R(A)); MAR2 = RA                     02738000 0
07B7  4809 0C41 2C0C 00F0          B L = A2                                                             02739000 0
07B8  000C 0000 0000 0020          COMP 16 = SAR            % (R(A)) IN UHW OF A2                       02740000 0
07B9  4849 00C1 0B00 00F0          1 L = B; SET LC2         % DECREMENT VALUE IN BIT 16                 02741000 0
07BA  4809 CC5E 0030 00F0          A2 - B = MIR                                                         02742000 0
07BB  7BC9 00C0 0000 00F0          IF ADV THEN SET LC1      % CHECK IF CARRY GENERATED                  02743000 0
07BC  1E70 0000 0030 0060          CARRY - 1 = CPCR                                                     02744000 0
07BD  4F10 0000 C000 0060          CHECKOV - 1 = CPCR       % CHECK IF OV GENERATED                     02745000 0
07BE  4809 0000 0000 C0F0          BMI                                                                  02746000 0
07BF  4809 0C40 8B30 00F0          B R = B,MIR              % RESULT INTO LHW OF B,MIR                  02747000 0
07C0  0000 0000 0C30 0020          16 = SAR                                                             02748000 0
07C1  2000 C000 0030 0060          SETCCA - 1 = CPCR                                                    02749000 0
07C2  2F5C 00C0 0000 0060          EOUTPUT - 1 = CPCR                                                   02750000 0
07C3  66EC 0000 0000 0040          OPCODE - 1 = MPCR                                                    02751000 0
                                                            %                                           02752000 0
                          OP02012:                          % INCREASE RA BY TWO; TYPE RR               02753000 0
                                                            % (R(A)) + 2 -> RA                          02754000 0
                                                            % B = (R(A)); MAR2 = RA                     02755000 0
07C4  4809 0C41 2030 00F0          B L = A2                                                             02756000 0
07C5  0020 0000 0000 00AC          COMP 16 = SAR; 2 = LIT% (R(A)) IN UHW OF A2                          02757000 0
```

```
07C6  4809 2001 0B30 00F0          LIT L = 0                                                  02758000 D
07C7  4809 CC40 0C30 00F0          A2 + B = MIR                                               02759000 D
07C8  78C9 0000 0C3C 00F0          IF AOV THEN SET LC1    % CHECK FOR CARRY                   02760000 D
07C9  1E7C 0003 0000 0060          CARRY - 1 = CPCR                                           02761000 D
07CA  4F1C 0003 0C30 0060          CHECKOV - 1 = CPCR     % CHECK FOR OVERFLOW                02762000 D
07CB  4809 0000 CC9C 00FC          BMI                                                        02763000 D
07CC  4809 0C40 8B9C 00F0          B R = B,MIR            % RESULT IN LHW OF B,MIR            02764000 D
07CD  000C 0000 0009 0020          16 = SAR                                                   02765000 D
07CE  200C 00C0 0090 006C          SETCCA - 1 = CPCR                                          02766000 D
07CF  2F5C 0003 000C 0060          EOUTPUT - 1 = CPCR                                         02767000 D
07D0  66EC 00C0 0090 0040          OPCODE - 1 = MPCR                                          02768000 D
                                                                                              02769000 D
                                   OP02D13:               % DECREASE RA BY TWO; TYPE RR       02770000 D
                                                          % (RA) - 2 -> RA                    02771000 D
                                                          % B = (RA); MAR2 = RA               02772000 D
07D1  4809 0C4L 2C9C C0F0          B L = A2                                                   02773000 D
07D2  0020 0000 0090 C0A0          COMP 16 = SAR; 2 = LIT% (R(A)) IN UHW OF A2                02774000 D
07D3  4849 2001 0B30 00F0          LIT L = 9; SET LC2     % LC2 IS SUBTRACTION FLAG           02775000 D
07D4  4809 CC5E 0090 C0F0          A2 - B = MIR                                               02776000 D
07D5  78C9 0000 C00C 00F0          IF AOV THEN SET LC1                                        02777000 D
07D6  1E7C 0000 0C00 006C          CARRY - 1 = CPCR       % SET/CLEAR CARRY BIT               02778000 D
07D7  4F1C 00C0 0C00 C060          CHECKOV - 1 = CPCR     % CHECK FOR OVERFLOW                02779000 D
07D8  4809 00C0 C090 C0F0          BMI                                                        02780000 D
07D9  4809 0C40 8B30 00F0          B R = B,MIR            % RESULT IN LHW OF B AND MIR        02781000 D
07DA  000C 0000 0090 0020          16 = SAR                                                   02782000 D
07DB  200D 00C0 0090 0060          SETCCA - 1 = CPCR                                          02783000 D
07DC  2F50 00C3 0090 C060          EOUTPUT - 1 = CPCR                                         02784000 D
07DD  66E0 0CC0 0030 004C          OPCODE - 1 = MPCR                                          02785000 D
                                                                                              02786000 D
                                   OP021:                 % LOAD DOUBLE; TYPE RI(II)          02787000 D
                                                          %(Y",Y"+1) -> RA,RA+1; SET CC       02788000 D
07DE  4809 2C56 0B3D 00F0          B AND LIT = B          % ISOLATE 'M' FIELD                 02789000 D
07DF  00F0 0000 0090 00E0          15 = LIT                                                   02790000 D
07E0  51CC 0000 0090 0060          REGSTACK - 1 = CPCR                                        02791000 D
07E1  2F3C 0003 0000 0060          EINPUT - 1 = CPCR      % B = (R(M)) = Y"                   02792000 D
07E2  482C 0003 0030 0060          LOBLE - 1 = CPCR       % CALL LOAD DOUBLE ROUTINE          02793000 D
07E3  66E0 0000 0C00 0040          OPCODE - 1 = MPCR                                          02794000 D
                                                                                              02795000 D
                                   OP023:                 % LOAD DOUBLE; TYPE RX              02796000 D
                                                          % (Y,Y+1) -> RA,RA+1                02797000 D
07E4  570C 0000 0000 0060          RXMFIELD - 1 = CPCR    % B = Y                             02798000 D
07E5  4820 0003 0C90 006C          LOBLE - 1 = CPCR       % CALL LOAD DOUBLE ROUTINE          02799000 D
07E6  56E0 0003 0C30 C04C          OPCODE - 1 = MPCR                                          02800000 D
                                                                                              02801000 D
                                   OPCODE03:              % THIS ROUTINE ANALYZES THE 03 OPCODE 02802000 D
07E7  5F9C 0000 0000 C060          XFCODE - 1 = CPCR                                          02803000 D
07E8  4809 C640 0040 00FC          A2 + AMPCR = AMPCR                                         02804000 D
07E9  11BC 0003 0030 00CC          OP03F - 1 = AMPCR                                          02805000 D
07EA  4809 00C3 0C00 00F0          STEP                                                       02806000 D
07EB  4824 00C0 0030 C0F0          EXEC                                                       02807000 D
                                                                                              02808000 D
07EC  11C0 00C0 0000 0040          OP03F:  OP030 - 1 = MPCR                                   02809000 D
07ED  3000 00C3 0C3C 0040          FAULT - 1 = MPCR                                           02810000 D
07EE  3000 0CC0 0090 0040          FAULT - 1 = MPCR                                           02811000 D
07EF  1190 C0C3 0030 C040          OP033 - 1 = MPCR                                           02812000 D
                                   OP030:                                                     02813000 D
07F0  4809 2C56 2030 00F0          B AND LIT = A2                                             02814000 D
07F1  00FC 0000 0C30 00E0          15 = LIT                                                   02815000 D
07F2  4809 C640 0030 00FC          A2 + AMPCR = AMPCR                                         02816000 D
07F3  11EC 0C03 0C3C 00CC          OP030M - 1 = AMPCR                                         02817000 D
```

```
07F4  4809 C15E 0000 C0F0          A2 GEQ LIT                                    02818000 D
07F5  0070 0000 0000 00E0          7 = LIT                                       02819000 D
07F6  7800 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP                   02820000 D
07F7  4E50 0000 0C30 0040          NOTIMP - 1 = MPCR      % INSTRUCTION NOT IMPLEMENTED  02821000 D
07F8  4824 0000 0000 00F0          EXEC                                          02822000 D
                                   %                                             02823000 D
                                   %                                             02824000 D
                             OP030H:                                             02825000 D
07F9  11F0 0000 0000 C040          OP03000 - 1 = MPCR                            02826000 D
07FA  1200 00C0 0C30 004C          OP03001 - 1 = MPCR                            02827000 D
07FB  1210 00C0 000C 0040          OP03002 - 1 = MPCR                            02828000 D
07FC  4E50 0CC0 0000 004C          NOTIMP - 1 = MPCR                             02829000 D
07FD  1220 0000 0030 0040          OP03004 - 1 = MPCR                            02830000 D
07FE  1230 0000 0C00 0040          OP03C05 - 1 = MPCR                            02831000 D
07FF  1240 0000 0000 0040          OP03006 - 1 = MPCR                            02832000 D
                                   %                                             02833000 D
                             OP03000:              % EXECUTIVE RETURN (P) + 1 -> RAJ HALT  02834000 D
0800  4809 E000 8B30 00F0          A3 R = B                                      02835000 D
0801  80F0 00C0 0030 C080          4 = SARJ 15 = LIT                             02836000 D
0802  4809 2C56 0B30 00F0          LIT AND B = B          % B = IA: FIELD        02837000 D
0803  51C0 0000 0C30 0060          REGSTACK - 1 = CPCR    % RA = MAR2            02838000 D
0804  4809 A0C1 0B30 00F0          A1 + 1 L = B           % ISOLATE PAR INTO B INCREMENTED BY 1  02839000 D
0805  0000 00C0 0000 0020          COMP 16 = SAR                                 02840000 D
0806  4809 0C40 8B30 00F0          B R = B,MIR            % (P) + 1 INTO B AND MIR  02841000 D
0807  20D0 00C0 0C30 C060          SETCCA - 1 = CPCR      % SET CONDITION CODE   02842000 D
0808  2F50 0000 0000 C060          EOUTPUT - 1 = CPCR     % (P) + 1 -> RA        02843000 D
0809  5D60 00C0 0000 0040          STOPTIME - 1 = MPCR    % STOP MACHINE EXECUTION  02844000 D
                                   %                                             02845000 D
                                   %                                             02846000 D
                             OP03001:              % STORE (SR1) INTO RA          02847000 D
080A  4809 E0C0 9000 C0F0          A3 R = A3                                     02848000 D
080B  80F0 00C0 0030 008C          4 = SARJ 15 = LIT                             02849000 D
080C  4809 E156 0B30 00F0          A3 AND LIT = B                                02850000 D
080D  51C0 0000 0C30 0060          REGSTACK - 1 = CPCR    % RA = MAR2            02851000 D
080E  4809 A0C0 8B30 00F0          A1 R = B,MIR           % (SR1) INTO MIR       02852000 D
080F  0000 00C0 0C30 0020          16 = SAR                                      02853000 D
0810  20D0 00C0 000C 0060          SETCCA - 1 = CPCR                             02854000 D
0811  2F50 00C0 0C00 0060          EOUTPUT - 1 = CPCR     % STORE (SR1) INTO RA  02855000 D
0812  66E0 0000 000C 0040          OPCODE - 1 = MPCR                             02856000 D
                                   %                                             02857000 D
                             OP03002:              % STORE (SR2) INTO RA          02858000 D
0813  4809 E0C0 9030 00F0          A3 R = A3                                     02859000 D
0814  80F0 00C0 0000 C080          4 = SARJ 15 = LIT                             02860000 D
0815  4809 E156 0B30 00F0          A3 AND LIT = B                                02861000 D
0816  51C0 00C0 0000 0060          REGSTACK - 1 = CPCR    % RA = MAR2            02862000 D
0817  4809 0000 0C30 24F0          ASE                                           02863000 D
0818  48C9 0F40 2030 C0F0          BMAR = A2              % RA INTO A2           02864000 D
0819  4809 20C3 001C 00FC          LIT = MAR2                                    02865000 D
081A  0220 0000 0030 00E0          STATUS2 = LIT                                 02866000 D
081B  2F30 00C0 0000 C060          EINPUT - 1 = CPCR      % D = SR2              02867000 D
081C  4809 0C41 0B30 C0F0          B L = B                % ISOLATE UYK2C SR2     02868000 D
081D  3000 00C3 0C30 C020          COMP 16 = SAR                                 02869000 D
081E  4809 0C40 8C30 00F0          B R = MIR                                     02870000 D
081F  4809 C0C3 0C1C 00F0          A2 = MAR2              % TRANSFER RA INTO MAR2  02871000 D
0820  20D0 00C0 003C C060          SETCCA - 1 = CPCR                             02872000 D
0821  2F50 00C0 0030 0060          EOUTPUT - 1 = CPCR     % (SR2) INTO RA        02873000 D
0822  66E0 00C3 0030 0040          OPCODE - 1 = MPCR                             02874000 D
                                   %                                             02875000 D
                                   %                                             02876000 D
                             OP03004:              % LOAD PJ (R(A)) INTO P        02877000 D
```

```
0823  4809 E000 9000 00F0          A3 R = A3                                                    02878000 D
0824  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                            02879000 D
0825  4809 E156 0B30 00F0          A3 AND LIT = B                                               02880000 D
0826  51C0 0003 0C30 0060          REGSTACK - 1 = CPCR     % RA IN MAR2                         02881000 D
0827  2F30 0000 0000 0060          EINPUT - 1 = CPCR       % B = (R(A))                         02882000 D
0828  4809 A000 C030 0CF0          A1 R = A1                                                    02883000 D
0829  0000 00C0 0090 0020          16 = SAR                                                     02884000 D
082A  4809 A001 4000 00F0          A1 L = A1               % ZERO PAR                          02885000 D
082B  4809 AC5C 4030 00F0          A1 OR B = A1            % (R(A)) INTO P UPPER 16 BITS UNTOUCHED 02886000 D
082C  66E0 0000 0030 0040          OPCODE - 1 = MPCR                                            02887000 D
                                                           %                                    02888000 D
                                                           %                                    02889000 D
                                   OP03005:                % LOAD SR1 ; (R(A)) INTO SR1          02890000 D
082D  4809 E000 9030 00F0          A3 R = A3                                                    02891000 D
082E  80F0 0000 0030 0080          4 = SAR; 15 = LIT                                            02892000 D
082F  4809 E156 0B00 C0F0          A3 AND LIT = B                                               02893000 D
0830  51C0 0003 0030 C060          REGSTACK - 1 = CPCR     % RA IN MAR2                         02894000 D
0831  2F30 0000 0000 0660          EINPUT - 1 = CPCR       % B = (R(A))                         02895000 D
0832  4809 0C41 009C 00FC          B L = MIR               % STORE (R(A)) IN UHW OF MIR          02896000 D
0833  0000 0003 0030 0020          COMP 16 = SAR                                                02897000 D
0834  4809 2001 2C30 00F0          LIT L = A2              % CREATE BIT MASK FOR LOADER TOGGLES  02898000 D
0835  3070 0003 0000 0080          7 = LIT; COMP 29 = SAR                                       02899000 D
0836  4809 2001 CB00 C0F0          LIT L = B               % LOWER MASK OF SR1                   02900000 D
0837  07C0 00C0 C030 00A0          112 = LIT; COMP 16 = SAR                                     02901000 D
0838  4809 CC5C 0B00 C0F0          A2 OR B = B             % CREATE MASK ONES IN UHW OF B        02902000 D
0839  4809 0C42 2D00 C0F0          NOT B = A2, BMI         % MASK FOR (RA)                      02903000 D
083A  4809 CC56 009C 00FC          A2 AND B = MIR          % MASK OFF (RA), STORE IN MIR         02904000 D
083B  4809 C002 0B3C 00F0          NOT A2 = B              % MASK FOR A1 (PSW)                   02905000 D
083C  4809 AC56 2D00 00FC          A1 AND B = A2, BMI      % MASK OFF SAVED PART OF A1 INTO A2   02906000 D
083D  4809 CC5C CB00 00F0          A2 OR B = B             % CREATE NEW A1 VALUE FROM (R(A))     02907000 D
083E  4809 ACC1 4000 C0F0         A1 L = A1               % CLEAR UHW OF A1                     02908000 D
083F  0000 0000 0C30 C020          COMP 16 = SAR                                                02909000 D
0840  4809 A000 C030 00FC          A1 R = A1               % RESTORE A1                          02910000 D
0841  4809 AC5C 4030 00F0          A1 OR B = A1            % RESTORE A1 (ACTIVE PSW)             02911000 D
0842  66E0 0000 0030 C040          OPCODE - 1 = MPCR       % UNTOUCHED                          02912000 D
                                                           %                                    02913000 D
                                   OP03006:                % LOAD SR2; (R(A)) INTO SR2           02914000 D
0843  4809 E000 9000 00F0          A3 R = A3                                                    02915000 D
0844  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                            02916000 D
0845  4809 E156 0B30 00F0          A3 AND LIT = B                                               02917000 D
0846  51C0 00C0 0000 0C60          REGSTACK - 1 = CPCR     % RA = MAR2                          02918000 D
0847  2F30 0000 0C30 006C          EINPUT - 1 = CPCR       % B = (R(A))                         02919000 D
0848  4809 0C40 2C30 00F0          B = A2                  % (R(A)) = A2                        02920000 D
0849  4809 2003 0C1C 00F0          LIT = MAR2                                                   02921000 D
084A  0220 0000 009C 00E0          STATUS2 = LIT                                                02922000 D
084D  2F30 0000 0000 0060          EINPUT - 1 = CPCR       % B = (STATUS2)                      02923000 D
084C  4809 0C40 8B30 00F0          B R = B                 % ISOLATE UPPER 16 BITS OF STATUS2    02924000 D
084D  000C 0000 0000 0020          16 = SAR                                                     02925000 D
084E  4809 0C41 0B00 00F0          B L = B                                                      02926000 D
084F  4809 CC5C C08C C0F0          A2 OR B = MIR           %  NEW STATUS2 CREATED              02927000 D
0850  2F50 0000 0030 C060          EOUTPUT - 1 = CPCR      % (R(A)) INTO STATUS2                02928000 D
0851  56E0 0003 009C 0040          OPCODE - 1 = MPCR                                            02929000 D
                                                           %                                    02930000 D
                                                           %                                    02931000 D
                                   OP033:                  % LOAD MULTIPLE, TYPE RX             02932000 D
                                                           % (Y ... Y + M - A + 1) INTO RA ... RM 02933000 D
0852  5330 0003 0030 0060          IFETCH - 1 = CPCR       % B = (Y) = Y                        02934000 D
0853  4809 0C41 0B3C 00F0          B L = B                                                      02935000 D
0854  00F0 0003 0000 00A0          COMP 16 = SAR; 15 = LIT                                      02936000 D
0855  4809 EC5C 1000 00F0          A3 OR B = A3            % STORE Y IN UHW OF A3               02937000 D
```

```
0856  4809 E156 2000 00F0          A3 AND LIT = A2          % A2 = :M: FIELD                    02938C00 D
0857  4809 EC00 8D00 00F0          A3 R = B                                                     02939C00 D
0858  5000 00C0 0030 0000          4 = SAR                                                      02940C00 D
0859  4809 2C56 0630 C0F0          LIT AND B = B            % B = :A: FIELD                     02941C00 D
085A  4809 CC5E 0000 CCFC          A2 LSS B                 % IF M < A THEN ADD 16              02942C00 D
085B  0100 00C0 0C00 00E0          16 = LIT                                                     02943C00 D
085C  7000 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP                                  02944C00 D
085D  4809 C140 2030 00F0          A2 + LIT = A2                                                02945C00 D
085E  4809 CC5E 2000 00F0          A2 - B = A2              % SUBTRACT M FROM A                 02946C00 D
085F  4809 C012 0030 C0FC          A2 EQL 0                 % IF M = A THEN COUNTER = 0         02947C00 D
0860  6819 0000 0000 00F0          IF TRUE THEN SKIP                                            02948C00 D
0861  4809 C0C0 2C90 24FC          A2 + 1 = A2; ASE         % COMPUTE M - A + 1                 02949C00 D
0862  51C0 0000 0030 0060          REGSTACK - 1 = CPCR      % B HAS RA; MAR2 = RA               02950C00 D
0863  4809 0F41 0020 40F0          BMAR L = BR1,CSAR        % SAVE RA IN BR1                    02951C00 D
0864  0000 0003 0030 C030          COMP 8 = SAR                                                 02952C00 D
0865  4809 E003 9010 C0FC          A3 R = A3,BR2            % STORE Y IN BP2                    02953C00 D
0866  4809 E0C0 9030 00F0          A3 R = A3                % STORE Y IN A3; RIGHT JUSTIFIED     02954C00 D
0867  60E0 00C3 0C90 C060    LDMUL:EMULIN - 1 = CPCR       % LOOP RETURN; B = (Y)              02955C00 D
0868  4809 0C40 009C C0F0          B = MIR                  % MIR = (Y)                         02956D00 D
0869  4809 00C0 0C90 20F0          ASR                                                          02957C00 D
086A  4809 0F40 8B1C C0F0          BMAR R = B,MAR2          % RA INTO B FROM BR1                02958C00 D
086B  0000 00C0 009C 0010          8 = SAR                                                      02959D00 D
086C  2F50 00C3 0000 0060          EOUTPUT - 1 = CPCR       % (Y) INTO RA                      02960C00 D
086D  4809 0C46 0B50 C0F0          B + 1 = B                % INCREMENT RA                      02961C00 D
086E  51C0 00C0 0000 C060          REGSTACK - 1 = CPCR      % COMPUTE NEXT RA INTO MAR2        02962C00 D
086F  4809 0F41 0020 00F0          BMAR L = BR1             % RA + 1 = BR1                     02963C00 D
0870  0000 00C0 0030 0030          COMP 8 = SAR                                                 02964C00 D
0871  4809 E0C0 1090 C0F0          A3 + 1 = A3              % DR2 + 1 = BR2; INCREMENT Y        02965C00 D
0872  4809 E001 0010 00FC          A3 L = BR2                                                   02966C00 D
0873  4809 C0DE 2000 00FC          A2 - 1 = A2              % DECREMENT COUNTER                 02967C00 D
0874  4809 C012 0030 00FC          A2 EQL 0                                                     02968C00 D
0875  6819 0C00 0000 00F0          IF TRUE THEN SKIP                                            02969C00 D
0876  3660 C0F0 0C30 C040          LDMUL - 1 = MPCR                                             02970C00 D
0877  66EC 00C0 0C90 0040          OPCODE - 1 = MPCR                                            02971D00 D
                                                      %                                        02972C00 D
                                                      %                                        02973C00 D
                               OPCODE04:              % THIS ROUTINE ANALYZES THE 04 OPCODE    02974C0C D
0878  5F90 00C0 0C30 006C          XFCODE - 1 = CPCR        % :F: FIELD IN A2                   02975C00 D
0879  4809 C640 009C C0F0          A2 + AMPCR = AMPCR                                           02976C0C D
087A  1250 C00D 0090 00C0          OP04F - 1 = AMPCR                                            02977C00 D
087B  4809 C0C3 0C90 00F0          STEP                                                         02978C0C D
087C  4824 00C0 0C90 00F0          EXEC                                                         02979C00 D
                                                      %                                        02980C00 D
087D  1260 0003 009C C040    OP04F:OP040 - 1 = MPCR                                            02981C00 D
087E  3000 C000 0000 C040          FAULT - 1 = MPCR                                             02982C00 D
087F  3000 00C0 C09C C040          FAULT - 1 = MPCR                                             02983C09 D
0880  1270 0003 0C0C C040          OP043 - 1 = MPCR                                             02984C0C D
                                                      %                                        02985C00 D
                               OP040:                 % THIS ROUTINE ISOLATES THE :M: FIELD    02986C00 D
0881  4809 2C56 2000 00FC          LIT AND B = A2                                               02987C0C D
0882  00FC 0000 0000 00EC          15 = LIT                                                     02988C00 D
0883  4809 C15E 009C C0F0          A2 GEQ LIT                                                   02989D00 D
0884  0040 00C0 C030 C0E0          4 = LIT                                                      02990C0C D
0885  780B 0CF0 009C 00FC          IF TRUE THEN STEP ELSE SKIP                                  02991C00 D
0886  3000 0003 C09C C040          FAULT - 1 = MPCR                                             02992D00 D
0887  4809 C640 0040 C0F0          A2 + AMPCR = AMPCR                                           02993C00 D
0888  128C 00C0 0C00 00C0          OP04M - 1 = AMPCP                                            02994C00 D
0889  4809 0C03 0000 00F0          STEP                                                         02995C0C D
088A  4824 00C0 009C C0F0          EXEC                                                         02996C0C D
                                                      %                                        02997C00 D
```

```
088C  12A0 0003 0000 C040          OP0401 - 1 = MPCR                                  02999C00 D
088D  12B0 0003 000C 004C          OP04C2 - 1 = MPCR                                  03000C00 D
088E  12CC 0003 0030 0040          OP0403 - 1 = MPCR                                  03001C00 D
                                   %                                                  03002C00 D
                          OP0400:  % SQUARE ROOT                                      03003C00 D
                                   % SQRT(R(A),R(A+1)) = R(A+1); REM=R(A)             03004C00 D
                                   % THIS ROUTINE CALCULATES THE SQUARE ROOT          03005C00 D
                                   % OF A NUMBER PASSED VIA THE B REGISTER.          03006C00 D
                                   % THE SQUARE ROOT IS RETURNED IN A3,              03007000 D
                                   % REMAINDER IN A2.                                03008C00 D
                                   % SET OV AND CC BITS                              03009000 D
088F  4809 0C40 8B30 00FC          B R = B                                            03010C00 D
0890  80FC 0000 0000 008C          15 = LIT; 4 = SAR                                  03011C00 D
0891  4809 2C56 0B00 00F0          LIT AND B = B          % ISOLATE "A" FIELD         03012C00 D
0892  51CC 0003 0000 0060          REGSTACK - 1 = CPCR    % R(A) INTO MAR2            03013CC0 D
0893  2F30 0000 0000 006C          EINPUT - 1 = CPCR      % (R(A)) INTO B             03014C00 D
0894  4809 0C41 2030 00FC          B L = A2                                           03015C0C D
0895  0000 0000 0000 C020          COMP 16 = SAR                                      03016C00 D
0896  4809 0F47 0C20 00F0          BMAR + 1 L = BR1       % SAVE R(A+1) IN BR1         03017C00 D
0897  00C0 0000 0000 C03C          COMP 8 = SAR                                       03018CC0 D
0898  4809 0F46 0D30 00FC          BMAR + 1 = B           % R(A+1)                    0301900C D
0899  51CC 00C0 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2          0302CC00 D
089A  2F30 0000 0030 C06C          EINPUT - 1 = CPCR      % (R(A+1)) INTO B           03021000 D
089B  4809 CC5C 0B30 00FC          A2 OR B = B            % B = (R(A),R(A+1))         0302200C D
089C  4809 0C40 0C91 00F0          B = MIR, LCTR          % MIR USED AS TEMP STORAGE  03023C0C D
089D  A0F0 0000 0000 0000          30 = SAR; 15 = LIT     % CTR SET FOR 16 ITERATIONS 03024000 D
089E  4809 0003 C010 00F0          0 = BR2                                            03025CC0 D
089F  4809 C0C3 2C1C 00F0          C = A2,MAR2                                        03026C00 D
08A0  28C9 0003 0000 00F0  PAIR:   BMI; IF LC1                                        03027000 D
08A1  4809 0C40 9032 C0F0          B R = A3, INC          % FETCH NEXT MS PAIR OF DIGITS 03028C00 D
08A2  A000 0C00 0000 0030          30 = SAR                                           03029C00 D
08A3  4809 EF5C 1030 00F0          A3 OR BMAR = A3        % A3 = PREV. REM/NEXT PAIR OF DIGITS 03030C00 D
08A4  4809 0C41 0036 00F0          B L = MIR              % TEMP STORAGE              03031C00 D
08A5  A000 0003 0000 0030          COMP 2 = SAR           % SHIFT NEXT PAIR OF DIGITS INTO MS BITS 03032C0C D
08A6  8A4B 00C3 0C0C 00FC          IF COV THEN SET LC2; STEP ELSE SKIP                03033C00 D
08A7  120C 00C0 0C00 004C          SEND - 1 = MPCR        % FINISHED                  03034C00 D
08A8  300C 0000 0000 0030          COMP 1 = SAR                                       03035C00 D
08A9  4809 E012 0030 00F0          A3 EQL 0               % CHECK FOR 00 PAIR OF DIGITS 03036C00 D
08AA  6C0B C0C1 2C30 C0FC          IF TRUE THEN A2 L = A2; STEP ELSE SKIP             03037C00 D
08AB  89FC 00C0 0C00 C040          PAIR - 1 = MPCR        % EXAMINE NEXT PAIR OF DIGITS 03038C00 D
08AC  4809 C0C1 CB3C C0F0  SEND:   A2 L = B               % PARTIAL SORT EFFECTIVELY MULT BY 4 03039C00 D
08AD  300C 0000 0030 0030          COMP 1 = SAR                                       03040C0C D
08AE  4809 0C46 CB30 00FC          B + 1 = B                                          03041C00 D
08AF  48C9 EC5F 0C1C C0F0          A3 - B L = MAR2        % POTENTIAL REMAINDER       03042000 D
08B0  A00C 00C0 0000 0030          COMP 2 = SAR                                       03043C00 D
08B1  4F09 EC01 701C C0FC          IF MST THEN A3 L = MAR2; SET LC1; SKIP             03044000 D
08B2  4809 C0DC 2C00 C0FC          A2 OR 1 = A2           % BUILD PARTIAL SORT        03045C00 D
08B3  4809 CC01 2C00 0CF0          A2 L = A2              % PREPARE FOR NEXT DIGIT    03046CC0 D
08B4  B000 C0C3 0030 0030          COMP 1 = SAR                                       03047C00 D
08B5  3006 00C0 0030 00F0          IF NOT LC2 THEN STEP ELSE SKIP                     03048C0C D
08B6  89FC 0003 0000 C040          PAIR - 1 = MPCR                                    03049000 D
08B7  2C19 E000 0B00 00F0          IF LC1 THEN A3 = B; SKIP  % RESTORE REMAINDER INTO B 03050000 D
08B8  4809 0F40 8B30 00FC          BMAR R = B             % RESTORE REMAINDER INTO B  03051C0C D
08B9  2000 C0C0 0000 0000          2 = SAR                                            03052C00 D
08BA  4809 C0C0 9C30 00F0          A2 R = A3              % REQUIRED TO BALANCE LAST UNUSED L SHIFT 03053C00 D
08BB  1000 0000 0030 0000          1 = SAR                                            03054C0C D
08BC  4809 0C40 2000 0CF0          B = A2                 % REMAINDER INTO A2         03055C00 D
08BD  4809 E0C0 0B30 00FC          A3 = MIR,B                                         03056C00 D
08BE  2000 00C0 0C0C 006C          SETCCA - 1 = CPCR      % SET THE CONDITION BITS    03057000 D
```

```
08C0  4809 0F40 801C 00F0                  BMAR R = MAR2           % (RA+1)
08C1  000C 0000 0030 C010                  8 = SAR
08C2  2F50 0000 0C00 0060                  EOUTPUT - 1 = CPCR      % SORT INTO R(A+1)
08C3  4809 0F40 1C30 00F0                  BMAR = A3
08C4  4809 C00E 0C1C 00F0                  A3 - 1 = MAR2           % R(A)
08C5  4809 C000 0030 00F0                  A2 = MIR
08C6  2F5C 0000 C000 0060                  EOUTPUT - 1 = CPCR      % REMAINDER INTO R(A)
08C7  5050 00C0 0000 C060                  CLEAROV - 1 = CPCR
08C8  66E0 0000 0030 0040                  OPCODE - 1 = MPCR
                                                                   %
                                                                   %
                            OP0401:                                % RVR; REVERSE REGISTER. (REVERSE RA)
08C9  4809 0C40 8800 C0F0                  B R = B                 % :A: FIELD LS BITS OF B
08CA  80FC 0000 0030 0080                  4 = SAR; 15 = LIT
08CB  4809 2C56 0B00 00FC                  LIT AND B = B           % B = :A: FIELD
08CC  51C0 0000 003C 006C                  REGSTACK - 1 = CPCR     % RA = MAR2
08CD  2F30 00C0 0000 C060                  EINPUT - 1 = CPCR       % B = (R(A))
08CE  4809 CC40 100C 00F0                  B = A3
08CF  4809 0000 2C00 C0F0                  0 = A2                  % ZERO A2
08D0  00E0 00C0 0C00 C0E0                  14 = LIT
08D1  4812 00C0 0001 C0F0                  LCTR; SAVE              % EXECUTE LOOP 16 TIMES
08D2  4809 E0C0 0000 C0F0                  A3
08D3  5C19 C00C 2030 00F0                  IF LST THEN A2 OR 1 = A2; SKIP  % WRITE 1 INTO LSB OF A2
08D4  4809 C0CE 2000 C0F0                  A2 AND B110 = A2        % WRITE 0 INTO LSB OF A2
08D5  4809 E000 9C00 40F0                  A3 R = A3,CSAR          %  MOVE (R(A)) RIGHT 1 BIT
08D6  1C00 00C0 0C00 0000                  1 = SAR                 % BIT SHIFT VALUE
08D7  9429 CC01 2022 C0F0                  IF NOT COV THEN A2 L = A2,INC; JUMP
08D8  4809 C000 0B30 C0F0                  A2 = B,MIR              % A2 CONTAINS REVERSE REG.
08D9  2000 00C0 0030 0060                  SETCCA - 1 = CPCR
08DA  2F5C 0000 0C00 0060                  EOUTPUT - 1 = CPCR
08DB  66E0 00C0 0000 0040                  OPCODE - 1 = MPCR
                                                                   %
                            OP0402:                                % CNT; COUNT ONES; COUNT -> RA + 1
08DC  4809 0C40 8800 00F0                  B R = B                 % :A: FIELD LS BITS OF B
08DD  80F0 00C0 0030 0080                  4 = SAR; 15 = LIT
08DE  4809 2C55 0B30 00F0                  LIT AND B = B
08DF  51CC 00C0 0000 0060                  REGSTACK - 1 = CPCR     % RA=MAR2
08E0  2F3C 0000 0030 C060                  EINPUT - 1 = CPCR       % B = (R(A))
08E1  4809 C0C0 2C30 C0F0                  0 = A2,MIR              % ZERO COUNT REGISTER
08E2  00E0 0000 0000 00E0                  14 = LIT
08E3  4812 0000 C001 00F0                  LCTR; SAVE              % EXECUTE LOOP 16 TIMES
08E4  4809 0C40 0030 00F0                  B
08E5  5C09 C0C0 2030 00F0                  IF LST THEN A2 + 1 = A2,MIR  % INCREMENT COUNTER
08E6  4809 0C40 8800 00F0                  B R = B
08E7  100C 0000 0030 C000                  1 = SAR                 % SHIFT RA RIGHT
08E8  8429 0000 0072 00F0                  IF NOT COV THEN INC; JUMP
08E9  4809 CF46 0E30 00FC                  BMAR + 1 = B            % R(A+1)
08EA  51CC 00C0 0C30 0060                  REGSTACK - 1 = CPCR     % R(A+1) INTO MAR2
08EB  2F50 00C0 0C3C 0060                  EOUTPUT - 1 = CPCR      % STORE COUNT IN MIR INTO RA + 1
08EC  66EC C000 0030 C040                  OPCODE - 1 = MPCR
                                                                   %
                            OP0403:                                % SFR; SCALE FACTOR. (RA,RA+1) SHIFTED
                                                                   % LEFT UNTIL (R(A))15 NEQ (R(A))14
                                                                   % SHIFT COUNT -> RA+2
08ED  4809 0C40 8830 00FC                  B R = B
08EE  80FC 0000 000C C080                  4 = SAR; 15 = LIT
08EF  4809 2C56 0E00 C0F0                  LIT AND B = B           % :A: FIELD INTO B
08F0  51CC 0000 0030 C060                  REGSTACK - 1 = CPCR     % RA = MAR2
08F1  2F3C 0000 0000 0060                  EINPUT - 1 = CPCR       % B = (R(A))
```

```
08F2  4809 0C41 2C00 24F0          B L = A2; ASE                                  03118C00 D
08F3  0000 C00D 0000 0020          COMP 16 = SAR                                  03119C00 D
08F4  4809 0F41 0C20 00F0          BMAR L = BR1          % STORE RA INTO BR1      03120C00 D
08F5  0000 C000 0000 0030          COMP 8 = SAR                                   03121000 D
08F6  4809 0F46 C800 00F0          BMAR + 1 = B          % R(A+1)                 03122000 D
08F7  51CC 0000 00?C 0060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2       03123C00 D
08F8  2F3C 0000 0030 0060          EINPUT - 1 = CPCR     % B = (R(A+1))           03124C00 D
08F9  4809 20C0 C030 C0FC          LIT = MIR                                      03125C00 D
08FA  01F0 0000 0000 C0E0          31 = LIT                                       03126C00 D
08FB  4809 CC5C 2C20 24F0          A2 OR D = A2; ASE     % A2 = (RA,RA+1)         03127C00 D
08FC  4809 C0C0 0000 00F0          A2                                             03128C00 D
08FD  6419 C012 0030 00F0          IF NOT ABT THEN A2 EOL 0 ; SKIP                03129000 D
08FE  12E0 00C0 0000 0040          SFTSIP - 1 = MPCR     % STOP COUNT; COUNT = 31  03130C00 D
08FF  680B 0000 0000 00F0          IF TRUE THEN STEP ELSE SKIP                    03131000 D
0900  12F0 00C0 0030 0040          SFTSIP - 1 = MPCR     % STOP COUNT; COUNT = 31  03132000 D
0901  4809 18C1 8B30 00FC          0101 C = B            % MASK OF TWO BITS IN D   03133C00 D
0902  11E0 0000 0C00 0080          1 = SAR; 30 = LIT                              03134C00 D
0903  4809 00C0 C09C 00FC          0 = MAR2,MIR                                   03135C00 D
0904  4812 00C0 0031 00F0          LCTR; SAVE                                     03136C00 D
0905  4809 CC56 9030 00FC          A2 AND B R = A3                                03137C00 D
0906  A03C 0000 0000 00B0          30 = SAR; 3 = LIT                              03138C00 D
0907  4809 E012 0000 C0F0          A3 EOL 0              % TEST IF BOTH BITS EOL 0 03139C00 D
0908  6419 E152 0030 C0F0          IF FALSE THEN A3 EOL LIT; SKIP  % TEST IF BOTH BITS EOL 1  03140C00 D
0909  130C 00C0 0C3C 0040          SFTCNT - 1 = MPCR     % SHIFT COUNT INCREMENTED 03141C00 D
090A  6819 0003 0030 00FC          IF TRUE THEN SKIP                              03142C0C D
090B  1310 00C0 0C?0 C040          SFTSIP - 1 = MPCR     % STOP COUNT;  MIR AND B HAVE COUNT  03143C0C D
090C  4809 C0F1 2030 00F0  SFTCNT: A2 L = A2             % SHIFT (RA,RA+1) LEFT ONE POSITION  03144C00 D
090D  8000 0C00 0000 003C          COMP 1 = SAR                                   03145C00 D
090E  4809 0F45 003C C0FC          BMAR + 1 = MAR2,MIR                            03146C00 D
090F  8429 0000 0032 00F0          IF NOT COV THEN INC; JUMP                      03147C0C D
0910  4809 00C0 0030 2CF0  SFTSIP: ASR                                            03148C00 D
0911  4809 0F40 900C 00F0          BMAR R = A3           % BR1 INTO A3= RA        03149C00 D
0912  0C20 0000 0090 0090          8 = SAR; 2 = LIT                               0315CC0C D
0913  4809 E140 C00C 00F0          A3 + LIT = B          % B = R(A+2)             03151CC0 D
0914  51CC 00C3 C030 0060          REGSTACK - 1 = CPCR                            03152000 D
0915  2F5C 0000 C030 C060          EOUTPUT - 1 = CPCR    % COUNT INTO RA+2        03153C00 D
0916  66EC 00C3 0030 C040          OPCODE - 1 = MPCR                              03154C00 D
                                                         %                        03155C00 D
0917  4849 0000 0C30 00F0  OP043:                        % BLX; BYTE LOAD AND INDEX BY 1  03156C0C D
0918  570C 0CC0 0C30 C060          SET LC2               % SET BYTE FLAG           03157000 D
0919  4809 0C41 0C1C C0F0          B L = BR2                                      03158000 D
091A  0000 00C0 0000 C03C          COMP 8 = SAR                                   03159C30 D
091B  50E0 00C0 0C30 C060          EMULIN - 1 = CPCR     % B = (Y)                03161C00 D
091C  0FFC 0000 0000 0090          8 = SAR; 255 = LIT                             0316?000 D
091D  3C19 2C56 0090 00FC          IF LC2 THEN LIT AND B = MIR; SKIP  % LS BYTE OF Y  03163C0C D
091E  4809 0C40 809C C0FC          B R = MIR             % MS BYTE OF Y            03164C00 D
091F  1CBC 00C3 0CCC 0060          AEQM - 1 = CPCR       % TEST IF A EQUAL M; B="A", A2="M"  03165C0C D
0920  51C0 00C0 0C30 0060          REGSTACK - 1 = CPCR                            03166C00 D
0921  4809 C0C3 0D30 00FC          BMI                                            03167000 D
0922  2F50 0003 0000 0060          EOUTPUT - 1 = CPCR    % BYTE LOAD INTO RA      03168000 D
0923  2000 00C0 0C3C C06C          SETCCA - 1 = CPCR                              03169C00 D
0924  380B 0000 0C00 C0F0          IF LC2 THEN STEP ELSE SKIP  % LC2 SET BY AEQM ROUTINE  0317CC3D D
0925  66E0 0CC0 C030 0040          OPCODE - 1 = MPCR                              03171C0C D
0926  4809 C0C0 0B30 00F0          A2 = B                                         03172C0C D
0927  51C0 00C0 0000 C06C          REGSTACK - 1 = CPCR   % RM = MAR2              03173C0C D
0928  2F3C 0000 0C0C 0060          EINPUT - 1 = CPCR     % B = (R(M))             03174C00 D
0929  4809 0C46 003C 00FC          B + 1 = MIR           % MIR = (R(M)) + 1        03175C00 D
092A  2F50 00C3 0C30 C060          EOUTPUT - 1 = CPCR                             03176C0C D
092B  66EC 00C0 0C0C C060          OPCODE - 1 = CPCR                              03177C0C D
```

```
                                                    %                                                   03179000 D
                                OPCODE05:           % THIS ROUTINE ANALYZES THE 05 OPCODE                03180000 D
092C  5F90 0000 0000 0060               XFCODE - 1 = CPCR                                               03181000 D
092D  4809 C640 0040 00FF               A2 + AMPCR = AMPCR                                              03182000 D
092E  1320 0000 0030 00C0               OP05F - 1 = AMPCR                                               03183000 D
092F  4809 6000 0000 00F0               STEP                                                            03184000 D
0930  4824 0000 0000 00FF               EXEC                                                            03185000 D
                                                    %                                                   03186000 D
0931  1330 0003 0000 0040  OP05F:        OP050 - 1 = MPCR                                               03187000 D
0932  1340 0003 0030 0040               OP051 - 1 = MPCR                                                03188000 D
0933  3000 0000 0030 0040               FAULT - 1 = MPCR                                                03189000 D
0934  1350 0003 0000 0040               OP053 - 1 = MPCR                                                03190000 D
                                                    %                                                   03191000 D
                                OP050:              % SBR; SET BIT 1 -> (R(A))M                         03192000 D
0935  4809 0C40 88F0 60F0               B R = B                                                         03193000 D
0936  80F0 00C3 0000 0080               4 = SAR; 15 = LIT         % B = 'A' FIELD                       03194000 D
0937  4809 2C56 0B30 00F0               LIT AND B = B                                                   03195000 D
0938  51C0 0000 0030 C060               REGSTACK - 1 = CPCR       % MAR2 = RA                           03196000 D
0939  2F30 0003 0030 0060               EINPUT - 1 = CPCR         % B = (R(A))                          03197000 D
093A  4809 0C40 2C00 00F0               B = A2                                                          03198000 D
093B  4809 E156 0B00 00F0               A3 AND LIT = B            % B = 'M' FIELD                       03199000 D
093C  00F0 00C0 0000 00E0               15 = LIT                                                        03200000 D
093D  4809 0C5E 0000 80F0               - B = SAR                 % VARIABLE SHIFT AMOUNT FOR LEFT SHIFT 03201000 D
093E  4809 00C1 0B30 00F0               8001 L = B                                                      03202000 D
093F  4809 CC5C 0B3C 00F0               A2 OR B = MIR,B           % SET 1 INTO (R(A))M AND STORE INTO MIR 03203000 D
0940  2F50 00C0 0030 C060               EOUTPUT - 1 = CPCR        % MODIFIED (R(A)) -> RA                03204000 D
0941  2D0C 0003 0070 0060               SETCCA - 1 = CPCR                                               03205000 D
0942  66E0 0003 0000 0040               OPCODE - 1 = MPCR                                               03206000 D
                                                    %                                                   03207000 D
                                OP051:              % LXI; LOAD AND INDEX BY 1;                          03208000 D
                                                    % TYPE RI(2)                                        03209000 D
0943  4809 2C56 0B30 00F0               LIT AND B = B                                                   03210000 D
0944  00F0 0003 0000 00EC               15 = LIT                  % B = 'M' FIELD                       03211000 D
0945  51C0 0003 0030 0060               REGSTACK - 1 = CPCR       % MAR2 = RM                           03212000 D
0946  2F30 00C0 0000 0060               EINPUT - 1 = CPCR         % (R(M)) = Y# = B                     03213000 D
                                                    %                                                   03214000 D
0947  4809 0C41 CC10 C0F0  LDINX1:      B L = BR2                                                       03215000 D
0948  0C00 00F0 0000 C030               COMP B = SAR                                                    03216000 D
0949  60E0 0003 0000 0060               EMULIN - 1 = CPCR         % B = (Y#) OR (Y)                     03217000 D
094A  4809 0C40 0030 00F0               B = MIR                   % MIR = (Y#) OR (Y)                   03218000 D
094B  1C80 00C0 0000 006C               AEQM - 1 = CPCR           % IS A EQUAL M ; B='A', A2 = 'M'       03219000 D
094C  51C0 0000 0000 0060               REGSTACK - 1 = CPCR       % RA = MAR2                           03220000 D
094D  2F50 0000 0030 0060               EOUTPUT - 1 = CPCR        % (Y) OR (Y#) -> RA                   03221000 D
094E  4809 00C0 0D00 C0F0               BMI                                                             03222000 D
094F  20DC 0000 0000 0060               SETCCA - 1 = CPCR         % (Y) OR (Y#) -> B                    03223000 D
0950  380B C000 0030 00FC               IF LC2 THEN STEP ELSE SKIP  % LC2 SET IN AEQM                   03224000 D
0951  56E0 0000 0030 0040               OPCODE - 1 = MPCR                                               03225000 D
0952  4809 C000 CB00 00F0               A2 = B                                                          03226000 D
0953  51C0 0000 0000 0060               REGSTACK - 1 = CPCR                                             03227000 D
0954  2F3C 0000 0030 0060               EINPUT - 1 = CPCR         % B = (R(M))                          03228000 D
0955  4809 0C45 0030 00F0               B + 1 = MIR               % MIR = (R(M)) + 1                    03229000 D
0956  2F50 0000 0000 0060               EOUTPUT - 1 = CPCR                                              03230000 D
0957  56E0 0000 0000 0040               OPCODE - 1 = MPCR                                               03231000 D
                                                    %                                                   03232000 D
                                OP053:              % LX; LOAD AND INDEX BY 1                            03233000 D
                                                    % TYPE RX                                           03234000 D
0958  5700 0003 0090 C060               RXMFIELD - 1 = CPCR       % B = Y                               03235000 D
0959  9460 0003 0000 0040               LDINX1 - 1 = MPCR         % LOAD AND INDEX BY 1                 03236000 D
                                                    %                                                   03237000 D
```

```
                        OPCODE08:                       % THIS ROUTINE ANALYZES THE 08 OPCODE
095A  5F90 0009 0030 0060               XFCODE - 1 = CPCR
095B  4809 C640 0040 00F0               A2 + AMPCR = AMPCR
095C  1360 0000 0030 00C0               OP06F - 1 = AMPCR
095D  4809 0000 0000 00F0               STEP
095E  4824 00C0 003C 00FC               EXEC
                                        %
095F  137C 00C0 C000 C040       OP06F:  OP060 - 1 = MPCR
0960  1380 C000 0000 C040               OP061 - 1 = MPCR
0961  30C0 0000 003C C040               FAULT - 1 = MPCR
0962  139C 0000 00FC 0040               OP063 - 1 = MPCR
                                        %
                        OP060:                          % ZBR; ZERO BIT; C -> (R(A))M
0963  4809 0C40 8B30 00F0              0 R = B
0964  80F0 00C0 0030 C080              4=SAR; 15=LIT
0965  4809 2C56 0B00 00F0              LIT AND B = B               % B = :A: FIELD
0966  51CC 00C0 0030 0060              REGSTACK - 1 = CPCR
0967  2F30 C000 0000 006C              EINPUT - 1 = CPCR           % B = (R(A))
0968  4809 0C40 2000 C0F0              B = A2
0969  4809 E156 0B3C 00FC              A3 AND LIT = B              % B = :M: FIELD
096A  0CFC 0000 003C 00E0              15 = LIT
096B  4809 2C5E 0C30 80F0              LIT - B = SAR               % VARIABLE SHIFT AMOUNT FOR RIGHT SHIFT
096C  02C0 00C0 003C 00E0              32 = LIT
096D  4809 00C9 8B3C C0F0              B110 C = B                  % MASK
096E  4809 CC56 0B30 C0F0              A2 AND B = MIR,B            % C -> (R(A))M
096F  2F50 00C0 000C C06C              EOUTPUT - 1 = CPCR          % MODIFIED (R(A)) -> RA
0970  66EC 0C00 C000 0040              OPCODE - 1 = MPCR
                                        %
                        OP061:                          % LDXI; LOAD DOUBLE (INDIRECT)
                                                        % INDEX BY TWO; R1(2)
0971  4809 2C56 0B3C 00F0              LIT AND B = B
0972  00F0 00C0 0000 00EC              15 = LIT
0973  51C0 0000 0C30 0060              REGSTACK - 1 = CPCR
0974  2F30 0000 0C0C 0060              EINPUT - 1 = CPCR           % B = (R(M)) = Y*
0975  482C 00C0 0030 0060              LOBLE - 1 = CPCR
                                        %
0976  1CBC 00C3 003C 006C       LDINX2: AEOM - 1 = CPCR
0977  4809 C000 0B3C 00FC              A2 = B
0978  3806 00C0 C000 00F0              IF LC2 THEN STEP ELSE SKIP  % :A: = :M:
0979  66E0 00C0 0030 C040              OPCODE - 1 = MPCR
097A  51CC 00C0 0030 C060              REGSTACK - 1 = CPCR         % RM = MAR2
097B  2F30 0003 0000 0060              EINPUT - 1 = CPCR           % B = (R(M))
097C  4809 2C40 C030 C0F0              LIT + B = MIR
097D  0C2C 00C3 0000 00E0              2 = LIT
097E  2F50 00C0 0000 0060              EOUTPUT - 1 = CPCR          % (R(M)) + 2 -> RM
097F  66E0 00C0 0C00 0060              OPCODE - 1 = CPCR
                                        %
                        OP063:                          % LDX; LOAD DOUBLE, INDEX BY TWO
                                                        % TYPE RX
0980  5700 0000 003C C060              RXMFIELD - 1 = CPCR         % R = Y
0981  482C 00C0 003C 006C              LOBLE - 1 = CPCR            % LOAD DOUBLE
0982  9750 00C0 0030 0040              LDINX2 - 1 = MPCR           % INDEX BY TWO
                                        %
                                        %
                        OPCODE07:                       % THIS ROUTINE ANALYZES THE 07 OPCODE
0983  5F90 0009 0030 C060              XFCODE - 1 = CPCR
0984  4809 C640 0040 00F0              A2 + AMPCR = AMPCR
0985  13AC 00C0 003C 00C0              OP07F - 1 = AMPCR
0986  4809 00C0 0030 C0F0              STEP
0987  4824 00C0 00CC 00F0              EXEC
```

```
0988  1380 0000 0030 C040        OP07F:  OP070 - 1 = MPCR                                        03299C00 D
0989  1300 00C0 0000 C040                OP071 - 1 = MPCR                                        033C0C00 D
098A  3000 00C0 0030 0040                FAULT - 1 = MPCR                                        03301C00 D
098B  13D0 00C0 000C C040                OP073 - 1 = MPCR                                        03302C00 D
                                                                 %                               03303C00 D
                                  OP070:                         % CBR; COMPARE BIT; TEST BIT M OF RA FCR 03304C00 D
                                                                 % ZERO                          03305C00 D
098C  4809 0C40 8B00 C0FC                B R = B                                                 03306C00 D
098D  80F0 C0C3 CC0C 0080                4 = SAR; 15 = LIT                                       03307C00 D
098E  9809 2C55 0B30 C0F0                LIT AND B = B                                           03308C00 D
098F  51C0 0000 0000 C060                REGSTACK - 1 = CPCR     % RA = MAR2                     03309C00 D
0990  2F30 00C0 000C C060                EINPUT - 1 = CPCR       % B = (R(A))                    0331CC00 D
0991  4809 0C40 2030 C0F0                B = A2                                                  03311C00 D
0992  4809 E155 0B30 C0F0                A3 AND LIT = B          % ISOLATE :M: FIELD              03312C00 D
0993  00FC 00C0 0C0C 00EC                15 = LIT                                                03313C0C D
0994  4809 CC40 003C 80F0                B = SAR                 % SHIFT COMPARE BIT TO LS FOSITION 03314C00 D
0995  4809 C0C3 A00C C0FC                A2 R = A2                                               03315C00 D
0996  4809 C0C0 0CC0 00F0                A2                                                      03316C0C D
0997  580B 00C0 CC30 00F0                IF LST THEN STEP ELSE SKIP  % TEST BIT                  03317C00 D
0998  13E0 C0C3 003C 0040        TEST1:  TEST1 - 1 = MPCR        % TEST FOR 1 TRUE               03318C00 D
0999  9809 2C52 0000 00F0                LIT EQL B               %  BIT POSITION 15?             03319C00 D
099A  6C19 1B09 8B30 00FC                IF TRUE THEN B010 C = B; SKIP  % SET 00 INTO CC LITS    03320000 D
099B  9809 1B17 8B0C 00F0                B011 C = B              % SINGLE BIT SET TO 0            03321C0C D
099C  0C00 00C3 0C30 0010                B = SAR                                                 03322C00 D
099D  9809 AC56 4000 00F0                A1 AND B = A1           % SET 0 OR 00 INTO CC BITS       03323000 D
099E  66E0 C0C3 003C C040                OPCODE - 1 = MPCR                                       03324C00 D
                                                                 %                               03325C00 D
099F  9809 2C52 000C 00F0        TEST1:  LIT EQL B               % BIT POSITION 15?               03326C00 D
09A0  6C19 1BC1 8B30 0CF0                IF TRUE THEN B101 C = B; SKIP  % SET 11 IN BOTH LC BITS 03327C00 D
09A1  9809 1BC1 8B30 C0F0                B100 C = B              % SET 1 INTO LS BIT OF CC        03328000 D
09A2  0000 00C0 0030 0010                B = SAR                                                 03329000 D
09A3  9809 AC5C 4000 00F0                A1 OR B = A1            % SET 1 OR 11 INTO CC BIT        0332AC00 D
09A4  66E0 0000 C00C 0040                OPCODE - 1 = MPCR                                       0332B000 D
                                                                 %                               0332C000 D
                                  OP071:                         % LPI; LOAD PSW (INDIRECT)       0332CC00 D
                                                                 % (Y*,Y*+1,Y*+2) -> P,SR1,SR2   0332D000 D
09A5  9809 2C56 0B0C 00F0                LIT AND B = B                                           0332E000 D
09A6  00F0 00C3 0C30 00E0                15 = LIT                                                0332EC00 D
09A7  51C0 0000 0C30 0060                REGSTACK - 1 = CPCR     % RM = MAR2                     0332F000 D
09A8  2F3C 00C0 0C30 0060                EINPUT - 1 = CPCR       % B = (R(M)) = Y*               0332FC00 D
09A9  4809 0C40 203C 00FC        LPSW:   B = A2                  % STORE B IN A2                  03330C00 D
09AA  9809 0C41 001C 00FC                B L = BR2               % Y* = BR2 OR Y = BR2            03331000 D
09AB  000C 00C3 0C0C C030                COMP 8 = SAR                                            03331C00 D
09AC  60E0 0000 0C00 0060                EMULIN - 1 = CPCR       % B = (Y*) OR (Y)               03332CC0 D
09AD  4809 A0C0 C000 00F0                A1 R = A1                                               03333C00 D
09AE  000C 00C3 0C30 C020                16 = SAR                                                03334000 D
09AF  4809 A0C1 9C00 00F0                A1 L = A1                                               03334C0C D
09B0  4809 AC5C 4020 00FC                A1 OR B = A1            % PAR= (Y) OR (Y*)              03335C00 D
09B1  4809 C0C0 2C00 00F0                A2 + 1 = A2             % NEXT SEQ Y OR Y*              03336C00 D
09B2  4809 C001 0C10 C0F0                A2 L = BR2                                              03337000 D
09B3  0000 0C00 0C3C C030                COMP 8 = SAR                                            03337C00 D
09B4  60EF 0000 0030 C060                EMULIN - 1 = CPCR       % (Y*+1) OR (Y+1) = B           03338C00 D
09B5  4809 0C41 0C3C 00F0                B L = MIR               % STORE B INTO UHW OF MIR        03339C00 D
09B6  0000 C000 000C 0020                COMP 16 = SAR                                           0333AC00 D
09B7  4809 2001 1000 00F0                LIT L = A3              % CREATE BIT MASK FOR LOADER TOGGLES 0333B000 D
09B8  3070 00C3 002C 0080                7 = LIT; COMP 29 = SAR                                  0333B00C D
09B9  4809 20C1 0B0C C0F0                LIT L = B               % STORE MASK FOR LOWER SR1 INTC B 0333CC00 D
09BA  0700 0003 0C3C 00A0                112 = LIT; COMP 16 = SAR                                0333D00C D
09BB  4809 EC5C 0B0C 00F0                A3 OR B = B             % CREATE FULL WORD MASK OF 1:S   0333E000 D
```

```
09BC  4809 0C42 1000 00F0          NOT D = A3, BMI        % MASK FOR (RA)                  03356CN0 D
09BD  4809 EC56 0030 0CF0          A3 AND B = MIR         % MASK OFF (RA), STORE IN MIR    03359000 D
09BE  4809 E002 0E30 00F0          NOT A3 = D             % MASK FOR A1 (PSW)              03360000 D
09BF  4809 AC56 1D00 00FC          A1 AND B = A3, BMI     % MASK OFF SAVED PART OF A1 INTO A2 03361000 D
09C0  4809 EC5C 0B30 C0F0          A3 OR D = B            % CREATE NEW A1 VALUE FROM (R(A)) 03362000 D
09C1  4009 A001 9000 00F0         A1 L = A1              % CLEAR UHW OF A1                 03363000 D
09C2  0000 CCC3 0030 0020          COMP 16 = SAR                                           03364CC0 D
09C3  4809 A000 C000 C0EC          A1 R = A1              % RESTORE A1                     03365000 D
09C4  4809 AC5C 9000 00F0          A1 OR B = A1                                           03366000 D
09C5  4809 CCC0 2030 00F0          A2 + 1 = A2            % NEXT SEQ Y OR Y*               03367CC0 D
09C6  4809 C001 0C10 00FC          A2 L = BR2                                             03368000 D
09C7  0000 00C0 0030 0030          COMP 8 = SAR                                           03369C00 D
09C8  60E0 0000 0C30 0060          EMULIN - 1 = CPCR      % (Y*+2) OR (Y+2) = B           0337CC00 D
09C9  4809 0C40 2000 00F0          D = A2                                                 03371000 D
09CA  4809 2003 001C 00F0          LIT = MAR2             % SR2 = MAR2                    03372000 D
09CB  0220 00C0 0030 C0E0          STATUS2 = LIT                                          03373000 D
09CC  2F30 00C3 0000 0060          EINPUT - 1 = CPCR      % D = (STATUS2)                 03374000 D
09CD  4809 0C40 8B30 C0F0          D R = B                % ISOLATE UPPER 16 BITS OF STATUS2 03375000 D
09CE  0000 00C0 0030 C020          16 = SAR                                               03376000 D
09CF  4809 0C41 0B00 00FC          B L = B                                                03377C00 D
09D0  4809 CC5C 0030 00F0          A2 OR B = MIR          % CREATE NEW STATUS2             03378C0C D
09D1  2F50 0003 0000 0060          EOUTPUT - 1 = CPCR                                     03379C00 D
09D2  66E0 C0C0 C000 0040          OPCODE - 1 = MPCR                                      0338CC00 D
                                                          %                               03381C00 D
                              OP073:                      % LPS LOAD PSW                  03382000 D
                                                          % (Y,Y+1,Y+2) -> P,SR1,SR2      03383C00 D
09D3  5700 00C3 0000 0060          RXMFIELD - 1 = CPCR    % B = Y                         03384C00 D
09D4  9A80 00C0 0000 0040          LPSW - 1 = MPCR                                        03385C00 D
                                                          %                               03386CC0 D
                                                          %                               03387C00 D
                              OPCODE10:                   % LOGICAL RIGHT SHIFT            03388000 D
09D5  5F90 00C0 0000 0060          XFCODE - 1 = CPCR      % RETURNS "F" IN A2              03389C00 D
09D6  4809 C640 0030 C0F0          A2 + AMPCR = AMPCR                                     03390C00 D
09D7  13F0 C003 0C0C C0C0          OP10F - 1 = AMPCR                                      03391C00 D
09D8  4809 0000 0C30 C0F0          STEP                                                   03392C00 D
09D9  4824 0CC3 0C30 C0F0          EXEC                                                   03393000 D
                                                          %                               03394CC0 D
                                                          %                               03395000 D
09DA  1400 0000 C000 0040     OP10F:  OP100 - 1 = MPCR                                    03396000 D
09DB  3000 0000 0C00 0040          FAULT - 1 = MPCR                                       03397000 D
09DC  1410 00C0 0000 0040          OP102 - 1 = MPCR                                       03398000 D
09DD  1420 0000 0C0C 0040          OP103 - 1 = MPCR                                       03399C00 D
                                                          %                               039C0000 D
                                                          %                               03401C00 D
                              OP100:                      % RR TYPE LOGICAL RIGHT SHIFT    03402000 D
                                                          % SHIFT (R(A)) RIGHT BY THE LS 5 BITS 03403000 D
                                                          % IN (R(M)), ZERO FILL, AND SET CC 03404C00 D
09DE  2380 0003 CC3C 0060          CONTENTSRM - 1 = CPCR   % (R(M)) INTO B                 03405CC0 D
09DF  4809 0C40 0030 00F0          E = MIR                % MIR USED AS TEMP FOR (R(M))    03406C00 D
09E0  4809 E0C3 0B30 C0F0          A3 = B                                                 03407CC0 D
09E1  2280 0003 0000 0060          CONTENTSRA - 1 = CPCR   % (R(A)) INTO B, R(A) IN MS WORD OF A3 03408C00 D
09E2  4809 0C43 1D30 00FC          B = A3, BMI                                            03409C00 D
09E3  4809 0C40 C030 80F0          B = SAR                % SAR HOLDS BITS 0-5 OF (R(M)) 03410C00 D
09E4  4809 E0C0 8B8C 0CF0          A3 R = MIR, B                                          03411CC0 D
09E5  20D0 0CC3 0C30 006C          SETCCA - 1 = CPCR      % SET CONDITION BITS            03412C00 D
09E6  2F50 00C3 0000 0060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A))              03413000 D
09E7  66E0 00C0 0030 0040          OPCODE - 1 = MPCR                                      03414C00 D
                                                          %                               03415C00 D
                                                          %                               03416C00 D
                              OP102:                      % RK TYPE LOGICAL RIGHT SINGLE SHIFT 03417C00 D
```

```
                                                        % SHIFT (R(A)) RIGHT Y BITS 0-5 PLACES
                                                        % ZERO FILL AND SET CC
09E8  6330 0000 0030 C060               IFETCH - 1 = CPCR       % Y INTO B
09E9  4809 0C40 0030 C0F0               B = MIR
09EA  4809 E156 0B30 C0F0               A3 AND LIT = B          % ISOLATE "M" FIELD
09EB  00F0 0000 0C00 C0E0               15 = LIT
09EC  4809 0C52 0C00 00F0               C EOL B
09ED  6C19 0000 0B00 00F0               IF TRUE THEN 0 = DJ SKIP
09EE  238C 0000 0000 C06C               CONTENTSRM - 1 = CPCR  % (R(M)) INTO B
09EF  4809 0C40 2D00 00F0               B = A2, BMI             % TRANSFER (R(M)) INTO A2
09F0  4809 CC40 0C90 00F0               A2 + B = MIR            % Y INTO MIR
09F1  4809 E0C0 0B0C C0F0               A3 = B                  % INSTRUCTION INTO B
09F2  22BC 00C0 000C 006C               CONTENTSRA - 1 = CPCR % (R(A)) INTO B
09F3  4809 0C40 1D30 00FC               B = A3, BMI             % (R(A)) INTO A3
09F4  4809 0C40 0030 80F0               B = SAR                 % Y(0-5) INTO SAR
09F5  4809 E0C0 883C 0CF0               A3 R = MIR, B           % PERFORM SHIFT
09F6  2F5C 0000 0C0C 0060               EOUTPUT - 1 = CPCR
09F7  20DC 0000 003C C060               SETCCA - 1 = CPCR       % SET APPROPRIATE CONDITION BITS
09F8  66E0 0000 000C 0040               OPCODE - 1 = MPCR
                                        %
                                        %
                                OP103:                          % RX TYPE BYTE STORE
                                                                % (R(A)) BITS 0-7 INTO Y BYTE
09F9  4849 00C0 003C C0F0               SET LC2                 % LC2 USED AS BYTE OPERATION FLAG
09FA  5700 0003 0C3C 006C               RXMFIELD - 1 = CPCR     % Y ADDR INTO B
09FB  4809 0C41 0010 00F0               B L = BR2               % Y INTO BR2
09FC  0000 0003 0C3C 003C               COMP 8 = SAR
09FD  4809 0F40 0020 C0F0               EMAR = BR1              % STORE Y
09FE  60EC 00C3 0000 0060               EMULIN - 1 = CPCR       % (Y) INTO B
09FF  4809 0C40 0C3C 00F0               B = MIR                 % (Y) INTO MIR
0A00  4809 E0C0 C03C C0F0               A3 = B                  % RESTORE INSTRUCTION INTO B
0A01  22BC 0003 0020 C060               CONTENTSRA - 1 = CPCR  % (R(A)) INTO B
0A02  4809 2C57 1D0C C0F0               LIT AND B L = A3, BMI  % (R(A)) BITS (0-7) INTO 2ND LSB OF A3
0A03  0FFC 0003 000C 00BC               255 = LIT, COMP 8 = SAR % (Y) INTO B
0A04  380B 00C3 0000 0CF0               IF LC2 THEN STEP ELSE SKIP  % IF LC2 SET, PUT BYTE INTO
0A05  143C 0003 0C3C 0040               LS103 - 1 = MPCR         % LS BYTE OF Y
0A06  4809 0C41 0F00 40F0               B L = B, CSAR           % CLEAR THE MS BYTE OF B
0A07  0C00 0000 0030 0010               COMP 24 = SAR
0A08  4809 0C40 8B00 00FC               B R = B
0A09  4809 E000 90C0 40F0               A3 R = A3, CSAR         % PUT (R(A)) -BITS 0-7 INTO 2ND LS BYTE
0A0A  0000 00C0 0000 001C               8 = SAR                 % IN A3
0A0B  4809 E0C1 1030 C0F0               A3 L = A3
0A0C  4809 EC5C 0C80 C0F0               A3 OR B = MIR           % MIR CONTAINS NEW (Y)
0A0D  1440 0000 0C00 0040               CNT103 - 1 = MPCR       % USED AS A "GO TO"
                                LS1C3:                          % (R(A)) BITS 0-7 INTO LS BYTE OF A3
0A0E  4809 0C40 8B00 40F0               B R = B, CSAR           % CLEAR LS BYTE OF B
0A0F  000C 00C0 0030 001C               8 = SAR
0A10  4809 0C41 0B3C 40FC               B L = B, CSAR
0A11  4809 E000 903C 00F0               A3 R = A3               % (R(A)) BITS 0-7 INTO LS BYTE OF A3
0A12  4809 EC5C 0C80 C0F0               A3 OR B = MIR           % MIR CONTAINS NEW (Y)
0A13  4809 0000 0070 20F0       CNT103: ASR                     % REFERENCE BR1
0A14  4809 0F40 0010 00F0               EMAR = BR2              % Y INTO BR2
0A15  61EC 00C0 C000 0060               EMULOUT - 1 = CPCR      % WRITE INTO R(A)
0A16  66E0 C0C0 0C00 004C               OPCODE - 1 = MPCR
                                        %
                                        %
                                OPCODE11:                       % RIGHT SHIFT AND STORE
0A17  5F90 0000 0C3C C060               XFCODE - 1 = CPCR       % "F" INTO A2
0A18  4809 C640 0030 00F0               A2 + AMPCR = AMPCR
0A19  145C 00C0 0000 C0C0               OP11F - 1 = AMPCR
```

```
0A1A  4809 0000 0030 00F0          STEP                                  03478C00 D
0A1B  4824 0000 C030 00FC          EXEC                                  03479C0C D
                                                        %                03480C00 D
                                                        %                03481000 D
0A1C  146C 0000 0030 0040    OP11F:  OP110 - 1 = MPCR                    03482C00 D
0A1D  147C 0000 0000 0040            OP111 - 1 = MPCR                    03483000 D
0A1E  148C 0000 0030 0040            OP112 - 1 = MPCR                    03484C00 D
0A1F  1490 0000 0030 0040            OP113 - 1 = MPCR                    03485C00 C
                                                        %                03486C0C D
                                                        %                03487000 D
                             OP110:                     % RR TYPE ALG RIGHT SINGLE SHIFT     03488000 D
                                                        % SHIFT R(A) RIGHT (R(M)) (0-5)      03489C00 D
                                                        % SIGN FILL AND SET CC               03490C00 D
0A20  2380 0000 0000 0060            CONTENTSRM - 1 = CPCR % (R(M)) INTO B                   03491C00 D
0A21  4809 0C40 0030 00F0            B = MIR             % TEMP STORAGE                      03492C00 D
0A22  4809 EC00 0B0C 00F0            A3 = B              % PUT INSTR BACK INTO B             03493C00 D
0A23  2280 0CC0 0000 0060            CONTENTSRA - 1 = CPCR % (R(A)) INTO B                   03494C00 D
0A24  4809 0C40 1000 00FC            B = A3, BMI         % (R(A)) INTO A3, (R(M)) INTO B     03495C00 D
0A25  4809 E0C1 9000 40F0            A3 C = A3, CSAR     % POS. SIGN BIT OF (R(A)) INTO LS BIT 03496C00 D
0A26  8000 00C0 0000 0010            15 = SAR                              03497C0C D
0A27  48C9 E000 0000 00F0            A3                                    03498C00 D
0A28  58C9 0000 0000 C0F0            IF LST THEN SET LC1 % FLAG FOR A NEG. (R(A))        03499C00 C
0A29  4809 E0C1 9000 00F0            A3 C = A3           % RESTORE A3                    03500C00 D
0A2A  4809 0C40 2000 00FC            B = A2              % A2 CONTAINS (R(M))            03501C00 D
0A2B  4809 0002 0B10 00F0            NOT 0 = B           % FILL B WITH ALL 1:S           03502C00 D
0A2C  4809 0C41 0B0C 00F0            B L = B             % B = 1111/0000                 03503C00 D
0A2D  000C 0003 0C0C 0020            16 = SAR                              03504C00 D
0A2E  2C09 EC5C 1000 00F0            IF LC1 THEN A3 OR B = A3              03505000 D
0A2F  4809 C000 0030 80F0            A2 = SAR            % SAR CONTAINS (R(M)) (0-5)     03506C00 D
0A30  4809 E000 8B00 C0F0            A3 R = B            % PERFORM ALG SHIFT             03507CC0 D
0A31  4809 0C41 0B70 C0F0            B L = B             % CLEAR UHW B                   03508C00 D
0A32  0000 0003 0030 0C20            COMP 16 = SAR                         03509C00 D
0A33  4809 0C40 8B30 00F0            D R = B, MIR        % RESTORE (R(A))                03510CC0 D
0A34  2F50 0003 0070 0060            EOUTPUT - 1 = CPCR  % WRITE NEW (R(A))             03511000 D
0A35  20D0 0000 C000 006C            SETCCA - 1= CPCR    % ARTHMETIC CC SETTING (ENTER WITH B) 03512C00 D
0A36  66E0 00C0 0030 0040            OPCODE - 1 = MPCR                     03513C00 D
                                                        %                03514C00 D
                                                        %                03515000 D
                             OP111:                     % RI TYPE 2 STORE, (R(A)) INTO Y*    03516C00 D
0A37  2280 0003 0C00 C060            CONTENTSRA - 1 = CPCR % (R(A)) INTO B, R(A) INTO MS A3  03517C00 C
0A38  4809 0C40 C030 00FC            B = MIR             % TEMP STORAGE OF (R(A))        03518C00 D
0A39  2380 0000 0000 0060            CONTENTSRM - 1 = CPCR % Y* INTO B                   03519C00 D
0A3A  4809 0C41 0010 00F0            B L = BR2                             03520C00 C
0A3B  000C CC00 0030 0030            COMP 8 = SAR                          03521C0C D
0A3C  61E0 CC00 0000 0060            EMULOUT - 1 = CPCR  % (R(A)) INTO Y*                03522C00 D
0A3D  66E0 0003 0030 0C40            OPCODE - 1 = MPCR                     03523000 D
                                                        %                03524C00 D
                                                        %                03525C0C D
                             OP112:                     % RK TYPE, ALG, RIGHT SINGLE SHIFT   03526C00 C
                                                        % SHIFT (R(A)) RIGHT Y (0-5)         03527C00 D
                                                        % SIGN FILL, AND SET CC              03528C00 C
0A3E  4809 2C56 0B30 00F0            LIT AND B = 6       % ISOLATE "M" FIELD              03529C00 D
0A3F  00F0 C000 0030 00EC            15 = LIT                              03530C00 D
0A40  4809 0C52 0C30 00FC            B EQL 0                               03531C00 D
0A41  6819 C000 0000 C0F0            IF TRUE THEN SKIF                     03532000 D
0A42  2380 0003 0030 0060            CONTENTSRM - 1 = CPCR  % (R(M)) INTO B              03533C00 C
0A43  4809 0C41 0B30 00F0            B L = B                               03534CC0 D
0A44  0C00 C003 0030 C020            COMP 16 = SAR                         03535C00 D
0A45  4809 EC5C 1000 C0F0            A3 OR B = A3        % A3 = (R(M))/INSTRUCTION       03536C00 D
0A46  6330 0000 0000 0060            IFETCH - 1 = CPCR   % Y INTO B                      03537090 D
```

```
0A47  4839 E0C0 AC00 C0F0          A3 R = A2                  % (R(M)) INTO A2                  03538C00 D
0A48  0000 00C0 0C00 0C2C          16 = SAR                                                     03539C00 D
0A49  4809 CC40 2000 00F0          A2 + B = A2                % Y INTO A2                        03540000 D
0A4A  4809 E0C0 9000 00F0          A3 R = A3                                                    03541C00 D
0A4B  80FC 00C3 0000 0080          4 = SAR; 15 = LIT                                            03542C00 D
0A4C  4809 E156 CB00 C0F0          A3 AND LIT = B             % ISOLATE "A" FIELD               03543C00 D
0A4D  51C0 00C0 0C00 006C          REGSTACK - 1 = CPCR        % ADDR OF R(A) INTO MAR2          03544C00 D
0A4E  2F30 C0C3 0C00 0060          EINPUT - 1 = CPCR          % (R(A)) INTO B                   03545C00 D
0A4F  4809 0C40 9000 00F0          B R = A3                   % PREPARE TO EXAMINE SIGN BIT     03546C00 D
0A50  8000 00C0 0000 CC10          15 = SAR                                                     03547C00 D
0A51  4809 E0C0 0000 00F0          A3                                                           03548C00 D
0A52  5C09 0002 1000 00F0          IF LST THEN NOT 0 = A3                                       03549C00 D
0A53  48F9 E3C1 1C00 00F0          A3 L = A3                  % A3 = 1111/0000                  03550C00 D
0A54  000C 0000 0C00 0020          COMP 16 = SAR                                                03551000 D
0A55  4809 EC5C 0B0C 00F0          A3 OR B = B                % P = 1111/(R(A)) OR 0000/(R(A))   03552C00 D
0A56  4809 C0C0 0000 80F0          A2 = SAR                   % Y (0-5) INTO SAR                03553C00 D
0A57  4809 0C40 8B00 00F0          B R = B                    % PERFORM ALG SHIFT                03554C00 D
0A58  4809 0C41 0B3C C0F0          B L = B                    % CLEAR UHW B                      03555C0C D
0A59  000C 0000 0C0C 0020          COMP 16 = SAR                                                03556C00 D
0A5A  4809 0C40 8B80 00F0          B R = B; MIR               % RESTORE SHIFTED (R(A))           03557C0C D
0A5B  2F50 C0C0 0000 0060          EOUTPUT - 1 = CPCR         % MAR2 HAS ADDR OF R(A)            03558C00 D
0A5C  2000 0000 0000 006C          SETCCA - 1 = CPCR          % SET THE CONDITION BITS          03559C00 D
0A5D  56EC 0000 0000 C060          OPCODE - 1 = CPCR                                            03560C00 D
                                                              %                                 03561C00 D
                                                              %                                 03562C00 D
                             OP113:                           % RX TYPE STORE, (R(A)) INTO Y     03563C0C D
0A5E  5700 00C0 0C3C 0060          RXMFIELD - 1 = CPCR        % B HOLDS ADDR OF Y                03564C00 D
0A5F  4809 0C41 0030 00F0          B L = MIR                  % Y INTO MIR                      03565C00 D
0A60  0C0C 0000 0C30 C03C          COMP B = SAR                                                 03566C0C D
0A61  4809 E0C3 9000 00F0          A3 R = A3                  % SHIFT "A" TO LS BITS             03567C0C D
0A62  80FC 00C0 0000 0080          4 = SAR; 15 = LIT                                            03568C00 D
0A63  4809 E156 CB30 00F0          A3 AND LIT = B             % "A" INTO B                      03569C00 D
0A64  51C0 0000 0C30 C060          REGSTACK - 1 = CPCR        % P(A) INTO MAR2                  03570C00 D
0A65  2F3C 0C00 0C00 C06C          EINPUT - 1 = CPCR          % (R(A)) INTO B                   03571C00 D
0A66  4809 0C40 0030 00F0          B = MIR;BMI                % (R(A)) INTO MIR                  03572C00 D
0A67  4809 0C43 0010 C0F0          B = BR2                    % Y INTO BR2                       03573C00 D
0A68  61EC 00C3 0C3C 0060          EMULOUT - 1 = CPCR         % (R(A)) INTO Y                    03574C00 D
0A69  66EC 0000 0030 0040          OPCODE - 1 = MPCR                                            03575C00 D
                                                              %                                 03576C00 D
                                                              %                                 03577C00 D
                             OPCODE12:                        % STORE AND RIGHT SHIFT            03578C00 D
0A6A  5F90 00C0 0000 C06C          XFCODE - 1 = CPCR          % "F" IN A2                        03579C00 D
0A6B  4809 C640 0030 00F0          A2 + AMPCR = AMPCR                                           03580C00 D
0A6C  14AC 0003 0C00 00C0          OP12F - 1 = AMPCR                                            03581000 D
0A6D  4809 0000 0C30 00F0          STEP                                                         03582C00 D
0A6E  4824 0003 000C 00F0          EXEC                                                         03583C00 D
                                                              %                                 03584C30 D
                                                              %                                 03585000 D
0A6F  1480 00C0 0000 0040  OP12F:  OP120 - 1 = MPCR                                           03586C00 D
0A70  14C0 00C0 0C30 0040          OP121 - 1 = MPCR                                           03587C0C D
0A71  14D0 C0C0 0C00 0040          OP122 - 1 = MPCR                                           03588C00 D
0A72  14EC 00C3 003C 0040          OP123 - 1 = MPCR                                           03589CC0 D
                                                              %                                 03590C00 D
                                                              %                                 03591C00 D
                             OP120:                           % RR TYPE LOGICAL RIGHT DOUBLE SHIFT  03592C00 D
                                                              % SHIFT R(A), R(A+1) RIGHT (R(M)) (0-5)  03593C00 D
                                                              % ZERO FILL AND SET CC             03594C00 D
0A73  4809 0C40 8B0C C0F0          B R = B                                                      03595C0C D
0A74  80F0 00C0 0C3C 0080          4 = SAR; 15 = LIT                                            03596C00 D
0A75  4809 2C56 0B30 C0F0          LIT AND B = B              % ISOLATE "A" FIELD               03597000 D
```

```
0A76  51C0 00C3 0000 0060          REGSTACK - 1 = CPCR     % ADDR OF R(A) INTO MAR      03598000 D
0A77  4809 0F41 0020 00F0          BMAR L = BR1            % SAVE R(A)                  03599000 D
0A78  0000 0000 0000 0030          COMP 0 = SAR                                         03600000 D
0A79  2F30 00C0 0000 C060          EINPUT - 1 = CPCR       % (R(A)) INTO B              03601000 D
0A7A  48C9 0C41 2000 00F0          B L = A2                % TEMP STORAGE OF (R(A))      03602000 D
0A7B  001C 0000 0000 C0AC          COMP 16 = SAR; 1 = LIT                               03603000 D
0A7C  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2      % R(A+1)                     03604000 D
0A7D  2F30 0003 0000 0060          EINPUT - 1 = CPCR       % (R(A+1)) INTO B            03605000 D
0A7E  4809 CC5C 2000 00F0          A2 OR B = A2            % A2 = (R(A))/(R(A+1))        03606000 D
0A7F  4809 E156 0B00 00F0          A3 AND LIT = B          % ISOLATE "M" FIELD           03607000 D
0A80  00F0 0000 0000 00FC          15 = LIT                                             03608000 D
0A81  51C0 0000 0000 0060          REGSTACK - 1 = CPCR     % R(M) INTO MAR2             03609000 D
0A82  2F30 0000 0000 0060          EINPUT - 1 = CPCR       % (R(M)) INTO B              03610000 D
0A83  48C9 0C40 0030 80F0          B = SAR                 % (R(M)) (C-5) INTO SAR       03611000 D
0A84  49C9 CP00 AB00 C0FC          A2 R = A2,B; SET LC1    % A2 HAS SHIFTED (R(A))/(R(A+1))  03612000 D
0A85  20DC 0000 0000 C060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS      03613000 D
0A86  4809 00C0 0000 20F0          ASR                     % REFEFENCE BR1               03614000 D
0A87  48C9 0F43 8C1C 00F0          BMAR R = MAR2           % ADDR OF R(A) INTO MAR2      03615000 D
0A88  000C 00C3 0030 0010          B = SAR                                              03616000 D
0A89  4609 C0C0 8030 00F0          A2 R = MIR              % (R(A)) INTO MIR             03617000 D
0A8A  0000 0000 0000 0020          16 = SAR                                             03618000 D
0A8B  2F5C 0000 0030 0060          EOUTPUT - 1 = CPCR      % WRITE OUT SHIFTED (R(A))   03619000 D
0A8C  4809 C001 2030 C0FC          A2 L = A2                                            03620000 D
0A8D  0010 0300 0030 00AC          COMP 16 = SAR; 1 = LIT                               03621000 D
0A8E  4809 CP00 8030 00F0          A2 R = MIR              % (R(A+1)) INTO MIR          03622000 D
0A8F  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2      % R(A+1)                     03623000 D
0A90  2F5C 0000 0030 0060          EOUTPUT - 1 = CPCR                                   03624000 D
0A91  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                    03625000 D
                                                           %                            03626000 D
                                                           %                            03627000 D
                              OP121;                       % RI TYPE 2 STORE DOUBLE      03628000 D
                                                           % (R(A),R(A+1)) INTO Y*, Y*+1 03629000 D
0A92  22B0 00C3 0C30 0060          CONTENTSRA - 1 = CPCR   % (R(A)) INTO B, R(A) INTO MS WORD A3  03630000 D
0A93  4809 0C43 0030 00FC          B = MIR                 % TEMP STORAGE FOR (R(A))     03631000 D
0A94  238C 00C3 0030 C060          CONTENTSRM - 1 = CPCR   % (R(M)) INTO B              03632000 D
0A95  48C9 0C41 0C10 C0F0          B L = BR2                                            03633000 D
0A96  0000 0000 0C30 C030          COMP 8 = SAR                                         03634000 D
0A97  61E0 00C3 0000 C060          EMULOUT - 1 = CPCR      % (R(A)) INTO Y*             03635000 D
0A98  4809 0F46 0C30 00FC          BMAR + 1 = MIR          % TEMP HOLDER FOR Y* + 1     03636000 D
0A99  48C9 E000 9000 40F0          A3 R = A3, CSAR                                      03637000 D
0A9A  0CF0 0000 0000 00A0          16 = SAR; 15 = LIT                                   03638000 D
0A9B  4809 E00C 001C 00F0          A3 OR 1 = MAR2          % R(A+1)                     03639000 D
0A9C  2F3C 00C3 0000 C060          EINPUT - 1 = CPCR       % (R(A+1)) INTO B            03640000 D
0A9D  4809 0C40 1000 00F0          B = A3                  % TEMP STORAGE OF (R(A))     03641000 D
0A9E  4809 00C0 0030 00F0          BMI                                                  03642000 D
0A9F  4809 E0C0 0C30 00F0          A3 = MIR                                             03643000 D
0AA0  4809 0C41 0C10 C0F0          B L = BR2               % ADDR OF Y* + 1              03644000 D
0AA1  000C 00C0 0C30 C030          COMP 8 = SAR                                         03645000 D
0AA2  61EC 00C0 0C30 C060          EMULOUT - 1 = CPCR                                   03646000 D
0AA3  66E0 0000 000C C040          OPCODE - 1 = MPCR                                    03647000 D
                                                           %                            03648000 D
                                                           %                            03649000 D
                              OP122;                       % RK TYPE LOGICAL RIGHT DOUBLE SHIFT  03650000 D
                                                           % SHIFT (R(A), R(A+1)) RIGHT Y(0-5)  03651000 D
                                                           % ZEROFILL AND SET CC        03652000 D
0AA4  633C 0000 00DC 0060          IFETCH - 1 = CPCR       % "Y" INTO B                  03653000 D
0AA5  4809 0C41 0B30 C0FC          B L = B                                              03654000 D
0AA6  00F0 00C0 0000 C0AC          16 = SAR; 15 = LIT                                   03655000 D
0AA7  4809 EC5C 1000 00F0          A3 OR B = A3            % A3 HOLDS "Y"/INSTRUCTION    03656000 D
0AA8  4809 E156 0B00 00F0          A3 AND LIT = B          % ISOLATE "M" FIELD          03657000 D
```

```
0AA9  4809 0C52 0000 00EC          B EOL 0
0AAA  6B09 00C0 0030 00F0          IF TRUE THEN SET LC1; SKIP ELSE STEP
0AAB  238C 00C0 0C30 0060          CONTENTSRM - 1 = CPCR   % (R(M)) INTO B
0AAC  4809 E000 A000 00F0          A3 R = A2               % "Y" INTO A2
0AAD  0000 0000 0000 002C          16 = SAR
0AAE  4809 CC40 2000 00F0          A2 + B = A2             % Y INTO A2
0AAF  4809 E0C0 8B00 0CFC          A3 R = B
0AB0  8000 0000 0000 000C          4 = SAR
0AB1  4809 2C56 0030 00F0          LIT AND B = B           % ISOLATE "A" FIELD
0AB2  00F0 00C0 0030 00E0          15 = LIT
0AB3  51CC 0003 0000 C060          REGSTACK - 1 = CPCR
0AB4  2F30 C0C3 0030 C060          EINPUT - 1 = CPCR       % (R(A)) INTO B
0AB5  4809 0F41 0020 00F0          BMAR L = BR1            % SAVE ADDR OF R(A) IN BR1
0AB6  0000 00C0 0C30 C030          COMP B = SAR
0AB7  4809 0C41 1000 00F0          6 L = A3                % (R(A)) INTO MS A3
0AB8  0010 0003 0000 00A0          16 = SAR; 1 = LIT
0AB9  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2      % R(A+1)
0ABA  2F3C 0003 0C00 0060          EINPUT - 1 = CPCR       % (R(A+1)) INTO D
0ABB  4809 EC5C 0B30 00F0          A3 OR B = B             % D = (R(A))/(R(A+1))
0ABC  4809 CD00 00DC E0F0          A2 = SAR                % Y (C-5) INTO SAR
0ABD  49C9 CC40 AB00 00F0          B R = B, A2; SET LC1    % PERFORM RIGHT SHIFT OF Y (C-5)
0ABE  20D0 00C3 0030 C060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS
0ABF  4809 C000 8C30 00F0          A2 R = MIR              % SHIFTED (R(A)) INTO MIR
0AC0  0000 00C3 0C00 0020          16 = SAR
0AC1  4809 00C0 0000 20F0          ASR                     % REFERENCE BR1
0AC2  4809 0F43 801C 00F0          BMAR R = MAR2           % ADDR OF R(A)
0AC3  0000 0003 0C00 C010          8 = SAR
0AC4  2F50 00C0 0030 C06C          EOUTPUT - 1 = CPCR      % WRITE NEW (R(A))
0AC5  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2      % R(A+1)
0AC6  001C 00C3 0030 00A0          1 = LIT; 16 = SAR
0AC7  4809 CD01 2C00 00FC          A2 L = A2               % ISOLATE (R(A+1))
0AC8  4809 CCC0 8C30 0CF0          A2 R = MIR
0AC9  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR        % WRITE NEW (R(A+1))
0ACA  66E0 00C3 0000 0040          OPCODE - 1 = MPCR
                                   %
                                   %
                                   % STORF DOUBLE
                                   % STORE (R(A)) INTO Y, (R(A+1)) INTO Y+1
0ACB  5700 0003 0030 0C60          RXMFIELD - 1 = CPCR      % Y INTO B
0ACC  4809 0C40 0030 00F0          B = MIR                 % TRANSFER Y TO MIR
0ACD  4809 E0C0 9000 00F0          A3 R = A3
0ACE  8000 0000 00C0 0000          4 = SAR
0ACF  4809 E156 0B30 C0F0          A3 AND LIT = B           % ISOLATE "A" FIELD
0AD0  00FC 0003 0030 00EC          15 = LIT
0AD1  51CC 00C0 0000 0060          REGSTACK - 1 = CPCR      % ADDR OF R(A) INTO MAR2
0AD2  2F30 00C0 0030 C060          EINPUT - 1 = CPCR        % (R(A)) INTO E
0AD3  4809 0C40 0030 C0F0          B = MIR,EMI              % (R(A)) INTO MIR, Y INTO E
0AD4  4809 0F40 1000 C0F0          BMAR = A3                % SAVE ADDR OF R(A) IN A3
0AD5  4809 0C41 0010 C0F0          B L = BR2
0AD6  000C 0000 003C 0030          COMP 8 = SAR
0AD7  61E0 0000 0000 0060          EMULOUT - 1 = CPCR       % WRITE (R(A)) INTO Y
0AD8  4809 2F5C 203C C0FC          LIT OR BMAR = A2         % Y + 1 ADDRESS
0AD9  0010 C0C0 0000 00E0          1 = LIT
0ADA  4809 E00C 001C C0F0          A3 OR 1 = MAR2           % ADDR OF R(A+1)
0ADB  2F30 0000 000C 0060          EINPUT - 1 = CPCR        % (R(A+1)) INTO B
0ADC  4809 0C40 0030 C0FC          B = MIR
0ADD  4809 C001 001C 00F0          A2 L = BR2               % ADDR OF Y + 1
0ADE  00FC C0C0 0000 003C          COMP 8 = SAR
0ADF  61E0 0000 0000 0060          EMULOUT - 1 = CPCR
0AE0  66E0 00C0 0000 0040          OPCODE - 1 = MPCR
```

```
                                                                      *
                                                                      *
                                    OPCODE13:                         * SHIFT
0AE1  5F90 0000 0000 0060                     XFCODE - 1 = CPCR       * "F" INTO A2
0AE2  4809 C640 0090 C0F0                     A2 + AMPCR = AMPCR
0AE3  14FC 0000 0000 C0C0                     OP13F - 1 = AMPCR
0AE4  4809 0000 0030 00FC                     STEP
0AE5  4824 0000 0030 00F0                     EXEC
                                                                      *
                                                                      *
0AE6  1500 0000 0000 0040           OP13F:    OP130 - 1 = MPCR
0AE7  3000 0000 0000 0040                     FAULT - 1 = MPCR
0AE8  1510 0000 0030 0040                     OP132 - 1 = MPCR
0AE9  1520 0000 0030 0040                     OP133 - 1 = MPCR
                                                                      *
                                                                      *
                                    OP130:                            * RR TYPE ALG RIGHT DOUBLE SHIFT
                                                                      * SHIFT (R(A),R(A+1)) RIGHT R(M) (0-5)
                                                                      * SIGN FILL AND SET CC
0AEA  4809 0C40 8B30 00F0                     B R = B
0AEB  80FC 0000 0030 0080                     4 = SAR; 15 = LIT
0AEC  4809 2C56 0B00 C0F0                     LIT AND B = B           * ISOLATE "A" FIELD
0AED  51CC 0000 0030 0060                     REGSTACK - 1 = CPCR     * ADDR OF R(A) INTO MAR2
0AEE  2F30 0000 0030 006C                     EINPUT - 1 = CPCR       * (R(A)) INTO B
0AEF  4809 0C41 2030 00F0                     B L = A2                * TEMP STORAGE OF (R(A))
0AF0  0010 0003 0000 C0A0                     COMP 16 = SAR; 1 = LIT
0AF1  4809 0F41 0020 00F0                     BMAR L = BR1            * TEMP STORAGE OF R(A)
0AF2  0000 00C3 0000 C030                     COMP 8 = SAR
0AF3  4809 2F5C 0C1C 00FC                     LIT OR BMAR = MAR2      * R(A+1) INTO MAR2
0AF4  2F30 00C0 000C C060                     EINPUT - 1 = CPCR       * (R(A+1)) INTO B
0AF5  4809 CC5C 2030 00F0                     A2 OR B = A2            * A2 = (R(A))/(R(A+1))
0AF6  4809 C000 0C30 C0FC                     A2
0AF7  4A49 0000 0C30 00F0                     IF MST THEN SET LC2
0AF8  4809 E156 0B00 0CF0                     A3 AND LIT = B          * ISOLATE "M"
0AF9  00F0 0003 0C0C 00E0                     15 = LIT
0AFA  51CC 0C03 0C0C C060                     REGSTACK - 1 = CPCR
0AFB  2F30 00C0 0C00 006C                     EINPUT - 1 = CPCR       * (R(M)) INTO B
0AFC  4809 0C40 0030 80F0                     B = SAR                 * (R(M)) (0-5) INTO SAR
0AFD  49C9 C0C0 AB00 00F0                     A2 R = A2,B; SET LC1    * A2 HAS RIGHT SHIFTED (R(A))/(R(A+1))
0AFE  20DC 0000 0000 C060                     SETCCA - 1 = CPCR       * SET THE CONDITION BITS
0AFF  3C19 0013 0B00 00F0                     IF LC2 THEN B111 L = B; SKIP
0B00  4809 0C00 0B0C C0F0                     0 = B
0B01  4809 CC5C 2030 20F0                     A2 OR B = A2; ASR
0B02  4809 0F43 801C 00F0                     BMAR R = MAR2           * ADDR OF R(A) INTO MAR2
0B03  0000 C000 0C30 0010                     8 = SAR
0B04  4809 C0C0 8030 C0F0                     A2 R = MIR              * (R(A)) INTO MIR
0B05  0000 0000 0000 C020                     16 = SAR
0B06  2F50 00C0 0030 0060                     EOUTPUT - 1 = CPCR      * WRITE OUT SHIFTED (R(A))
0B07  4809 C001 2030 00F0                     A2 L = A2
0B08  001C 0C03 0C0C 00AC                     COMP 16 = SAR; 1 = LIT
0B09  4809 C0C0 8830 00FC                     A2 R = MIR, B           * SHIFTED (R(A+1)) INTO MIR
0B0A  4809 2F5C 001C 00F0                     LIT OR BMAR = MAR2      * R(A+1)
0B0B  2F50 0000 0000 006C                     EOUTPUT - 1 = CPCR
0B0C  66E0 00C0 0030 0040                     OPCODE - 1 = MPCR
                                                                      *
                                                                      *
                                    OP132:                            * PK TYPE ALG RIGHT DOUBLE SHIFT
                                                                      * SHIFT (R(A),R(A+1)) RIGHT Y(0-5)
                                                                      * SIGN FILL AND SET CC
0B0D  4809 2C56 0B30 00F0                     LIT AND B = B
```

```
0F0E  00F0 0000 000C C0EC          15 = LIT
0F0F  4809 0C52 0000 00F0          B EQL 0                  % CHECK "M" FOR 0          03779C00 D
0F10  6819 0000 0C90 00F0          IF TRUE THEN SKIP;                                  03780000 D
0F11  2380 00C0 0C90 C06C          CONTENTSRM - 1 = CPCR    % (R(M)) INTO B            03781C00 D
0F12  4809 0C41 0B90 00F0          B L = B                                             03782C00 D
0F13  0000 0000 0C90 0020          16 = SAR                                            03783C00 D
0F14  4809 EC5C 1C9C 00F0          A3 OR B = A3             % A3 = (R(M))/INSTRUCTION   03784C0C D
0F15  6330 0000 0C90 0060          IFETCH - 1 = CPCR        % "Y" INTO B               03785000 D
0F16  4809 E0C0 AC90 00F0          A3 R = A2                % (R(M)) INTO A2            03786C00 D
0F17  000C 00C3 0090 C020          16 = SAR                                            03787C00 D
0F18  4809 CC40 2090 C0F0          A2 + B = A2              % Y INTO A2                 03788C00 D
0F19  4809 E0C0 8B90 00F0          A3 R = B                 % ISOLATE "A"               03789C00 D
0F1A  90FC C003 0C0C 0080          4 = SAR; 15 = LIT                                   0379CC00 D
0F1B  4809 2C56 0B30 00F0          LIT AND B = B            % "A" INTO B               03791C00 D
0F1C  51C0 00C3 0C9C C060          REGSTACK - 1 = CPCR      % ADDR OF R(A) INTO MAR2   03792C00 D
0F1D  2F30 0000 0090 006C          EINPUT - 1 = CPCR        % (R(A)) INTO B            03793C0C D
0F1E  4809 CC41 1C90 00F0          B L = A3                 % TEMP STORAGE OF (R(A))    03794C00 D
0F1F  0000 00C0 0090 0020          COMP 16 = SAR                                       03795C00 D
0F20  4809 CF41 0C20 C0F0          BMAR L = BR1             % SAVE ADDR OF R(A) IN BR1  03796C00 D
0F21  001C 0000 C090 00BC          COMP 8 = SAR ; 1 = LIT                              03797000 D
0F22  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2       % R(A+1)                   03798C00 D
0F23  2F30 00C0 0C9C 0060          EINPUT - 1 = CPCR        % (R(A+1)) INTO B          03799C00 D
0F24  4809 EC5C 1030 00F0          A3 OR B = A3             % A3 = (R(A)/(R(A+1))       03800C00 D
0F25  4809 E0C0 0070 00F0          A3                                                  03801C00 D
0F26  4A49 0000 0000 00FC          IF MST THEN SET LC2                                 0380 2C0C D
0F27  4809 C0C3 0C9C 80FC          A2 = SAR                 % Y(0-5) INTO SAR           0380 3C00 D
0F28  49C9 E003 9B0C 00F0          A3 R = A3,B; SET LC1     % SHIFT (R(A))/(R(A+1))     03804C00 D
0D29  2000 0000 0C90 0060          SETCCA - 1 = CPCR        % SET THE CONDITION BITS    03805C00 D
0E2A  3C19 0019 0B0C 00F0          IF LC2 THEN B111 L = B; SKIP                        0380 6000 D
0E2B  4009 0000 CB90 00F0          0 = B                                               0380 7000 D
0E2C  4809 EC5C 1C90 C0F0          A3 OR B = A3             % PERFORM SIGN FILL         03008C00 D
0E2D  4809 00C0 C00C 20F0          ASR                      % REFERENCE BR1            03809000 D
0E2E  4809 CF43 801C 00F0          BMAR R = MAR2            % ADDR OF R(A) INTO MAR2    0381 C000 D
0E2F  0000 00C0 0C90 C01C          8 = SAR                                             03811C00 D
0D30  48C9 E0C3 8030 00F0          A3 R = MIR                                          03812C00 D
0F31  00C0 0000 0000 002C          16 = SAR                                            03813C00 D
0F32  2F50 00C0 0090 C060          EOUTPUT - 1 = CPCR       % WRITE OUT NEW (R(A))     03814C00 D
0F33  0010 0000 0090 00A0          COMP 16 = SAR; 1 = LIT                              03815C00 D
0B34  4809 2F5C 001C 00FC          LIT OR BMAR = MAR2       % R(A+1)                   03816C00 D
0B35  4809 E001 1000 00F0          A3 L = A3                                           03817C00 D
0F36  4809 E0C0 8B90 00F0          A3 R = MIR, B            % NEW (R(A+1))              03818C00 D
0F37  2F50 0000 0C90 0060          EOUTPUT - 1 = CPCR                                  03819000 D
0F38  66E0 0000 0000 C040          OPCODE - 1 = MPCR                                   03820C00 D
                                                            %                          03821C00 D
                                                            %                          03822000 D
                                OP133:                      % RX TYPE STORE MULTIPLE    03823C00 D
                                                            % (R(A),...R(M)) INTO Y,...Y+M-A+1  03824C0C D
0F39  6330 0000 0000 0060          IFETCH - 1 = CPCR        % Y INTO B                  03825CCC D
0F3A  4809 0C41 C020 C0FC          B L = BR1                % SAVE Y IN BR1              03826C00 D
0F3B  0000 00C0 0C90 003C          COMP 8 = SAR                                        03827000 D
0B3C  4809 E156 2000 00FC          A3 AND LIT = A2          % "M" INTO A2               03828C00 D
0F3D  80FC 0003 0090 0080          4 = SAR; 15 = LIT                                   03829C00 D
0F3E  4809 E000 9C90 C0F0          A3 R = A3                                           03830C00 D
0B3F  4809 E155 0B90 00F0          A3 AND LIT = B, MIR      % "A" INTO B, MIR          03831000 D
0F40  4809 CC5E 0B3C 00F0          A2 - B = B               % CHECK IF "M" < "A"        03832C09 D
0F41  4419 0C46 0B00 C0F0          IF NOT MST THEN B + 1 = B; SKIP                     03833C00 D
0B42  4809 2C80 0B90 00F0          LIT + B = B              % CALCULATE 16 + (M-A) + 1  03834C30 D
0B43  0110 00C3 0000 00E0          17 = LIT                                            03835000 D
0B44  4809 0C5F 1000 C0F0          -B L = A3, BMI           % A3 IS NEG CTR, B HOLDS "A"  0383600C D
0B45  000C C0C0 0070 0020          COMP 16 = SAR                                       03837C0C D
```

```
0846  51C0 0000 0000 0060    STMU:   REGSTACK - 1 = CPCR    % ADDR OF R(A) INTO MAR2          03838000 D
0847  2F3C 00C0 0000 C060            EINPUT - 1 = CPCR     % (R(A)) INTO B                   03839C00 D
0848  4809 0F46 001C C0F0            BMAR + 1 = MAR2       % INCREMENT R(A)                  0384 0C00 D
0849  4809 E0C0 9000 00F0            A3 R = A3             % CLEAR LHW OF A3                 03841000 D
084A  0000 00C0 0000 0020            16 = SAR                                                03842000 D
084B  4809 E0C1 1000 00F0            A3 L = A3                                               03843F00 D
084C  4809 EF5C 1030 00F0            A3 OR BMAR = A3       % A3 = -CTR/R(A+X)               03844C00 0
084D  4809 0C40 0030 20FC            D = MIR, ASR                                            03845CC0 D
084E  4809 0F40 0010 00F0            BMAR = BR2            % LOAD BR2 WITH Y                 03846C00 D
084F  61EC 0000 0C70 0060            EMULOUT - 1 = CPCR    % WRITE OUT NEW Y VALUE           03847000 D
0850  4809 0F46 001C 00F0            BMAR + 1 = MAR2       % NEXT Y ADDRESS                  03848C00 D
0851  4809 0F41 0020 C0FC            BMAR L = BR1          % SAVE NEXT Y IN BR1              03849000 D
0852  000C 00C0 0030 0030            COMP B = SAR                                            03850C00 D
0853  4809 00C1 0830 00F0            1 L = B               % 1 IN BIT 16 OF B                03851000 D
0854  0000 0000 0030 0020            COMP 16 = SAR                                           03852C00 D
0855  4809 EC40 1000 00F0            A3 + B = A3           % INCREMENT NEGATIVE COUNTER       03853C00 D
0856  4809 E156 0B0C 00F0            A3 AND LIT = B        % ISOLATE ADDR OF NEXT R(A+X)     03854C00 D
0857  01F0 0000 0000 00E0            31 = LIT                                                03855CC0 0
0858  4809 E0C0 0030 00F0            A3                    % A3 TO THE ADDER                 03856C00 D
0859  480B 00C0 0030 00FC            IF MST THEN STEP ELSE SKIP  % IF CTR NEG, FETCH NEXT R(A+X)  03857C00 0
085A  845C 0000 0020 0040            STMU - 1 = MPCR       % RETURN TO TOP OF LOOP           03858C00 D
085B  56EC 00C0 00DC C040            OPC00E - 1 = MPCR                                       03859C00 D
                                                           %                                 03860C00 D
                                                           %                                 03861C00 0
                                     OPC00E14:             % ALGEBRAIC LEFT SHIFT             03862000 D
085C  5F9C 0000 0C30 0060            XFC00E - 1 = CPCR     % "F" INTO A2                     03863C00 D
085D  4809 C640 0080 C0FC            A2 + AMPCR = AMPCR                                      03864000 0
085E  153C 00C0 0030 00C0            OP14F - 1 = AMPCR                                       03865C00 0
085F  4809 00C0 0C9C 00F0            STEP                                                    03866C00 0
0860  4824 0000 000C 00F0            EXEC                                                    03867000 0
                                                           %                                 03868C00 D
                                                           %                                 03869C00 D
0861  1540 0C00 0020 0040    OP14F:  OP140 - 1 = MPCR                                        0387 0C0C D
0862  3000 00C0 0020 0040            FAULT - 1 = MPCR                                        03871C00 D
0863  1550 00C0 0C00 0040            OP142 - 1 = MPCR                                        03872000 D
0864  156C 0000 0000 C04C            OP143 - 1 = MPCR                                        03873C00 D
                                                           %                                 03874C00 D
                                                           %                                 03875C30 0
                                     OP140:                % RR TYPE ALG LEFT SHIFT (REGISTER)  03876C30 D
                                                           % SHIFT (R(A))  LEFT (R(M)) (0-5) PLACES  03877C00 D
                                                           % ZERO FILL, SET CC AND SET OVERFLOW BIT  03878C00 0
0865  2380 0003 0030 0060            CONTENTSRM - 1 = CPCR % (R(M)) INTO B                   03879C00 C
0866  4809 0C40 0030 C0F0            B = MIR               % TEMP STORAGE                     03880C00 0
0867  4809 E0C0 0B0C 00F0            A3 = B                % PUT INSTR BACK INTO B            03881C00 0
0868  22B0 00C3 0C0C 0060            CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                  03882C00 D
0869  4809 0C40 2030 00F0            0 = A2, BMI           % (R(A)) INTO A2, (R(M)) INTO B    03883000 D
086A  4809 2C56 0B3C 00F0            LIT AND B = B         % MASK BITS 0-5                    03884C0C 0
086B  03FC 0000 0030 C0E0            63 = LIT                                                03885C00 0
086C  4809 2C59 003C 00FC            LIT LEQ B             % CHECK FOR SHIFT >= 16            03886C00 0
086D  0100 0000 0C0C 0CE0            16 = LIT                                                03887C0C D
086E  7819 0000 C000 00F0            IF FALSE THEN SKIP                                      03888C00 D
086F  1570 C0C0 0030 0040            C140 - 1 = MPCR       % SHIFT >= 16                     03889C00 0
0870  4809 0C5E 003C 80FC            -B = SAR                                                0389FC00 D
0871  4809 C0C1 2000 C0F0            A2 L = A2             % PERFORM LEFT SHIFT               03891C00 D
0872  4809 C0C3 9C30 00FC            A2 R = A3             % PUT DIGITS SHIFTED OUT IN A3     03892C00 0
0873  000C 00C0 0030 0020            16 = SAR                                                03893C00 D
0874  4809 C001 203C C0FC            A2 L = A2             % CLEAR UHW A2                     03894C00 D
0875  4809 C0C0 8B30 C0FC            A2 R = MIR, B         % SHIFTED (R(A)) INTO MIR          03895C00 D
0876  4809 E0C2 0000 00FC            NOT A3                                                  03896000 D
0877  63C9 0000 0000 C0F0            IF NOT ABT THEN SET LC1  % FLAG FOR SETTING OV BIT      03897C00 D
```

```
0B78  5020 0000 0030 0060          OVBIT - 1 = CPCR       % SET OV BIT ACCORDINGLY          03898C00 D
0B79  2000 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS          03899C00 D
0B7A  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR                                       03900000 D
0B7B  66E0 0000 0030 0040          OPCODE - 1 = MPCR                                        03901000 D
0B7C  4809 C012 0030 00F0  C140:   A2 EQL 0              % CHECK FOR 0 IN R(A)             03902000 D
0B7D  63C9 0000 0000 00F0          IF FALSE THEN SET LC1                                    03903000 D
0B7E  5020 0000 0000 0060          OVBIT - 1 = CPCR       % SET OV BIT ACCORDINGLY          03904000 D
0B7F  4809 0000 0C30 00F0          0 = MIR                % SHIFT >= 16, -> 0  INTO R(A)     03905000 D
0B80  218C 0000 0C30 006C          SETOG - 1 = CPCR                                         03906000 D
0B81  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR                                       03907000 D
0B82  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                        03908000 D
                                                          %                                03909000 D
                                                          %                                03910000 D
                                   OP142:                 % RR TYPE ALG LEFT SINGLE SHIFT   03911000 D
                                                          % SHIFT (R(A)) LEFT Y (0-5) PLACES 03912000 D
                                                          % ZERO FILL AND SET CC, SET OVERFLOW BIT 03913000 D
0B83  6330 0000 0C30 0060          IFETCH - 1 = CPCR      % "Y" INTO B                      03914000 D
0B84  4809 0C40 0030 00FC          B = MIR                % "Y" INTO MIR                     03915000 D
0B85  4809 E156 0B30 00F0          A3 AND LIT = B         % ISOLATE "M"                     03916000 D
0B86  00FC 0000 0000 00EC          15 = LIT                                                 03917000 D
0B87  4809 0C52 0000 00F0          0 EQL B                % CHECK FOR 0 , "M" FIELD          03918000 D
0B88  6819 0000 0C70 C0F0          IF TRUE THEN SKIP                                        03919000 D
0B89  2380 0000 0000 006C          CONTENTSRM - 1 = CPCP  % (R(M)) INTO B                   03920000 D
0B8A  4809 0C40 2D30 00F0          B = A2, BMI            % (R(M) INTO A2, "Y" INTO B       03921000 D
0B8B  4809 CC40 0030 00F0          A2 + B = MIR           % Y INTO MIR                      03922000 D
0B8C  4809 EC00 9030 00F0          A3 R = A3                                                03923000 D
0B8D  80FC 0000 0C30 008C          4 = SAR; 15 = LIT                                        03924000 D
0B8E  48C9 E156 0B0C 00F0          A3 AND LIT = B         % ISOLATE "A"                     03925000 D
0B8F  51C0 00C0 0000 0060          REGSTACK - 1 = CPCR    % R(A) INTO MAR2                  03926000 D
0B90  2F30 C0C0 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B                   03927000 D
0B91  4809 0C40 2D30 C0F0          B = A2, BMI            % (R(A)) INTO A2, Y INTO 0         03928000 D
0B92  4809 2C56 CB0C 00F0          LIT AND B = B          % ISOLATE SHIFT AMOUNT             03929000 D
0B93  03F0 00C0 0000 00E0          63 = LIT                                                 03930000 D
0B94  4809 2C59 0030 00FC          LIT LEQ B              % IS SHIFT AMOUNT > 16 ?           03931000 D
0B95  0100 00C0 0030 00E0          16 = LIT                                                 03932000 D
0B96  7819 0000 0C0C 00F0          IF FALSE THEN SKIP                                       03933000 D
0B97  158C 0000 0000 004C          C142 - 1 = MFCR        % SHIFT > 16                      03934000 D
0B98  4809 0C55 0000 80F0          -B = SAR                                                 03935000 D
0B99  4809 CC01 2030 00FC          A2 L = A2              % PERFORM REQUIRED LEFT SHIFT      03936000 D
0B9A  4809 C000 9030 00F0          A2 R = A3              % PUT LEFT SHIFTED BITS INTO A3    03937000 D
0B9B  0C00 0000 0000 C020          16 = SAR                                                 03938000 D
0B9C  4809 C001 2030 00FC          A2 L = A2              % CLEAR UHW OF A2                  03939000 D
0B9D  4809 C0C0 8B30 00F0          A2 R = MIR, B          % NEW (R(A)) INTO MIR, B           03940000 D
0B9E  4809 E002 0C00 00FC          NOT A3                                                   03941000 D
0B9F  63C9 00C0 0000 C0FC          IF NOT ABT THEN SET LC1                                  03942000 D
0BA0  5020 00C0 CC3C 0060          OVBIT - 1 = CPCR       % SET THE OV BIT ACCORDINGLY      03943000 D
0BA1  2000 00C0 0000 0060          SETCCA - 1 = CPCR      % SET THE CC                      03944000 D
0BA2  2F50 0C00 0030 0060          EOUTPUT - 1 = CPCR     % WRITE OUT NEW R(A)              03945000 D
0BA3  66E0 0000 C00C 0040          OPCODE - 1 = MPCR                                        03946000 D
0BA4  4809 C012 0C3C 00F0  C142:   A2 EQL 0              % CHECK FOR A ZERO (R(A))          03947000 D
0BA5  63C9 0000 0000 00F0          IF FALSE THEN SET LC1                                    03948000 D
0BA6  502C C000 0C70 006C          OVBIT - 1 = CPCR       % SET THE OV BIT                  03949000 D
0BA7  4809 00C0 0C30 00FC          C = MIR                % PREPARE TO WRITE 0 INTO R(A)     03950000 D
0BA8  218C 00C0 C030 0060          SETOC - 1 = CPCR                                         03951000 D
0BA9  2F50 0000 C030 0060          EOUTPUT - 1 = CPCR                                       03952000 D
0BAA  66E0 0000 0C00 0040          OPCODE - 1 = MPCR                                        03953000 D
                                                          %                                03954000 D
                                                          %                                03955000 D
                                   OP143:                 % RX TYPE BYTE STORE INDEX        03956000 D
                                                          % (R(A)) BITS 0-7 INTO Y BYTE     03957000 D
```

```
0BA8  4849 0003 0030 00F0              SET LC2                  % BYTE INSTRUCTION FLAG          03959000 D
0BAC  5700 0000 0000 0060              RXMFIELD - 1 = CPCR      % Y ADDR INTO E                  0396C000 D
0BAD  4809 0C41 0030 00F0              D L = BR1                % TEMP STORAGE OF Y ADDR          03961000 D
0BAE  0000 0003 0000 0030              COMP 8 = SAR                                              03962000 D
0BAF  4809 E000 0600 00F0              A3 = 8                                                    03963000 D
0BB0  22B0 00C0 0070 0060              CONTENTSRA - 1 = CPCR    % (R(A)) INTO B                  03964000 D
0BB1  4809 2C55 0030 00F0              LIT AND B = MIR          % (R(A)) BITS 0-7 INTO MIR        03965000 D
0BB2  0FF0 0003 0030 00E0              255 = LIT                                                 03966000 D
0BB3  2380 0000 0030 0060              CONTENTSRM - 1 = CPCR    % (R(M)) INTO B                  03967000 D
0BB4  4809 0C45 0B30 00F0              D + 1 = B                                                 03968000 D
0BB5  4809 0C40 0D30 00F0              B = MIR, BMI             % (R(M)) + 1 INTO MIR, (R(A)) INTO B  03969000 D
0BB6  2F50 00C0 0000 0060              EOUTPUT - 1 = CPCR       % (R(M)) + 1 INTO R(M)           03970000 D
0BB7  4809 0C40 1030 00FC              B = A3                   % MOVE (R(A)) BITS 0-7 TO LS BYTE OF A3  03971000 D
0BB8  4809 C0C0 C030 20F0              ASR                      % REFERENCE BR1                   03972000 D
0BB9  4809 0F40 0010 00FC              BMAR = DP2               % Y ADDR INTO ER2                 03973000 D
0BBA  60EC 0000 0C30 0060              EMULIN - 1 = CPCR        % (Y) INTO B                      03974000 D
0BBB  3B0B 0FC0 0000 00F0              IF LC2 THEN STEP ELSE SKIP % IF LC2, PUT BYTE INTO LS BYTE  03975000 D
0BBC  159C 00C0 0C00 0060              LS143 - 1 = CPCR                                          03976000 D
0BBD  4809 0C41 0D30 40F0              B L = B, CSAR                                             03977000 D
0BBE  0000 00C0 0C30 C010              COMP 24 = SAR                                             03978000 D
0BBF  4809 0C40 8B00 00F0              B R = B                                                   03979000 D
0BC0  4809 E0C1 1000 00F0              A3 L = A3                                                 03980000 D
0BC1  0000 0000 0000 0030              COMP 8 = SAR                                              03981000 D
0BC2  4809 EC5C 0030 00F0              A3 OR B = MIR                                             03982000 D
0BC3  4809 0F41 0010 00F0              EMAR L = BR2             % SET UP BR2 FOR EMULOUT          03983000 D
0BC4  0000 00C0 0030 C030              COMP 8 = SAR                                              03984000 D
0BC5  61EC 00C0 0000 0060              EMULOUT - 1 = CPCR                                        03985000 D
0BC6  66E0 0000 0000 0040              OPCODE - 1 = MPCR                                         03986000 D
0BC7  4809 0C40 8B30 40F0      LS143:  B R = B, CSAR            % CLEAR LS BYTE OF B             03987000 D
0BC8  4809 0C41 0B00 00FC              B L = B                                                   03988000 D
0BC9  4809 EC5C 0C3C 00F0              A3 OR B = MIR                                             03989000 D
0BCA  4809 0F41 0010 00F0              EMAR L = BR2             % SET UP FOR EMULOUT              03990000 D
0BCB  61EC 0000 0C30 0060              EMULOUT - 1 = CPCR                                        03991000 D
0BCC  66E0 0000 0000 0040              OPCODE - 1 = MPCR                                         03992000 D
                                       %                                                         03993000 D
                                       %                                                         03994000 D
0BCD  5F90 0000 0000 0060      OPCODE15:  XFCODE - 1 = CPCR    % "F" INTO A2                    03995000 D
0BCE  4809 C640 0040 00F0              A2 + AMPCR = AMPCR                                        03996000 D
0BCF  15AC 00C0 0000 0CC0              OP15F - 1 = AMPCR                                         03997000 D
0BD0  4809 00C0 0030 00FC              STEP                                                      03998000 D
0BD1  4824 0000 000C 00FC              EXEC                                                      03999000 D
                                       %                                                         04000000 D
0BD2  15BC 00C0 C00C 0040      OP15F:  OP15C - 1 = MPCR                                         04001000 D
0BD3  15C0 0000 0C00 0040              OP151 - 1 = MPCR                                          04002000 D
0BD4  15D0 C000 003C 0040              OP152 - 1 = MPCR                                          04003000 D
0BD5  15E0 C000 0070 0040              OP153 - 1 = MPCR                                          04004000 D
                                       %                                                         04005000 D
                                       %                                                         04006000 D
                               OP150:                          % RR TYPE CIRCULAR SINGLE SHIFT   04007000 D
                                                               % SHIFT (R(A)) LEFT CIRCULARLY (R(M))  04008000 D
                                                               % BITS 0-5 AND SET CC             04009000 D
0BD6  2380 C000 003C 006C              CONTENTSRM - 1 = CPCR    % (R(M)) INTO B                  04010000 D
0BD7  4809 0C40 0030 00F0              B = MIR                  % (R(M)) INTO MIR                 04011000 D
0BD8  4809 EC00 0B30 00F0              A3 = B                                                    04012000 D
0BD9  22B0 0000 0C00 006C              CONTENTSRA - 1 = CPCR    % (R(A)) INTO B                  04013000 D
0BDA  4809 0C40 2000 00F0              E = A2                                                    04014000 D
0BDB  4809 C001 2030 00F0              A2 L = A2                                                 04015000 D
0BDC  03F0 00C3 C000 C0A0              COMP 16 = SAR, 63 = LIT                                   04016000 D
0BDD  4809 CC5C 2030 C0F0              A2 OR B = A2, BMI        % A2 = (R(A))/(R(A)), (R(M)) INTO E  04017000 D
```

```
0B0F  4812 0000 0000 00F0          SAVE                                                04019000 D
0BE0  4809 E15E 0000 00F0          A3 GEQ LIT                                          04020000 D
0BE1  0100 0000 0030 00E0          16 = LIT                                            04021000 D
0BE2  7C29 E15E 1000 00F0          IF TRUE THEN A3+ NOT LIT + 1 = A3; JUMP             04022000 D
0BE3  4809 EC00 0B00 00F0          A3 = B                                              04023000 D
0BE4  4809 2C5E 003C 80F0          LIT - B = SAR                                       04024000 D
0BE5  4809 C001 8000 00F0          A2 C = B                                            04025000 D
0BE6  4809 0C41 0B00 00F0          B L = B                                             04026000 D
0BE7  0000 0000 0000 C020          16 = SAR                                            04027000 D
0BE8  4809 0C40 883C 00F0          B R = MIR, B                                        04028000 D
0BE9  2F50 0000 0020 0060          EOUTPUT - 1 = CPCR                                  04029000 D
0BEA  2000 0000 000C C060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS     04030000 D
0BEB  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                   04031000 D
                                                          %                            04032000 D
                                                          %                            04033000 D
                          OP151:                          % RI TYPE 2 STORE AND INDEX BY 1   04034000 D
                                                          % (R(A)) INTO Y*, (R(M)) + 1 INTO R(M)  04035000 D
0BEC  2380 0000 003C 006C          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B              04036000 D
0BED  4809 0C41 002C 00F0          B L = BR1              % Y* INTO BR1                04037000 D
0BEE  0000 0000 0C0C 0030          COMP B = SAR                                        04038000 D
0BEF  4809 EC03 0830 00F0          A3 = B                                              04039000 D
0BF0  2280 0000 0000 C060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B              04040000 D
0BF1  4809 0C40 0C30 20F0          B = MIR; ASR                                        04041000 D
0BF2  4809 0F43 0010 C0F0          BMAR = BR2                                          04042000 D
0BF3  61EC 0000 0C00 006C          EMULOUT - 1 = CPCR     % (R(A)) INTO Y*             04043000 D
0BF4  4809 0F46 CC30 00FC          BMAR + 1 = MIR         % (R(M)) + 1                 04044000 D
0BF5  4809 E156 CB3C 00F0          A3 AND LIT = B         % ISOLATE "M" FIELD          04045000 D
0BF6  00F0 0000 0030 00E0          15 = LIT                                            04046000 D
0BF7  51CC 0000 0030 0060          REGSTACK - 1 = CPCR    % MAR2 = ADDR OF R(M)        04047000 D
0BF8  2F5C 0003 0C30 CC6C          EOUTPUT - 1 = CPCR     % (R(A)) + 1 INTO R(M)       04048000 D
0BF9  66EC 000D 0070 C040          OPCODE - 1 = MPCR                                   04049000 D
                                                          %                            04050000 D
                                                          %                            04051000 D
                          OP152:                          % RK TYPE CIRCULAR LEFT SINGLE SHIFT      04052000 D
                                                          % SHIFT (R(A)) Y (C-5) PLACES AND SET CC  04053000 D
0BFA  4809 2C56 083C C0F0          LIT AND B = B          % ISOLATE "M"                04054000 D
0BFB  00FC 000D 0030 00E0          15 = LIT                                            04055000 D
0BFC  4809 0C52 C03C 00F0          B EQL 0                                             04056000 D
0BFD  6819 0000 0C00 C0F0          IF TRUE THEN SKIP                                   04057000 D
0BFE  238C 0000 0030 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B              04058000 D
0BFF  4809 CC41 0B30 C0FC          B L = B                                             04059000 D
0C00  0000 0000 0C00 C02C          COMP 16 = SAR                                       04060000 D
0C01  4809 EC5C 1000 C0F0          A3 OR B = A3           % A3 = (R(M))/INSTRUCTION    04061000 D
0C02  633C 0000 0070 0060          IFETCH - 1 = CPCR      % Y INTO B                   04062000 D
0C03  4809 EC00 AC0C 0CF0          A3 R = A2              % (R(M)) INTO A2             04063000 D
0C04  000C 0000 C03C 0020          16 = SAR                                            04064000 D
0C05  4809 CC40 2000 C0FC          A2 + B = A2            % Y INTO A2                  04065000 D
0C06  4809 C156 0030 00FC          A2 AND LIT = MIR       % Y(C-5)                     04066000 D
0C07  93FC C000 0C0C 008C          63 = LIT; 4 = SAR                                   04067000 D
0C08  4809 E0C3 900C CGF0          A3 R = A3                                           04068000 D
0C09  4809 E156 083C 00F0          A3 AND LIT = B         % ISOLATE "A"                04069000 D
0C0A  00FC C000 0000 C0EC          15 = LIT                                            04070000 D
0C0B  51C0 0000 0C00 006C          REGSTACK - 1 = CPCR    % ADDR OF R(A) INTO MAR2     04071000 D
0C0C  2F30 0000 0C00 006C          EINPUT - 1 = CPCR      % (R(A)) INTO E              04072000 D
0C0D  4809 0C41 100C 00F0          B L = A3                                            04073000 D
0C0E  010C 0000 0C00 00A0          COMP 16 = SAR; 16 = LIT                             04074000 D
0C0F  4809 EC5C 2000 C0F0          A3 OR B = A2           % A2 = (R(A))/(R(A))         04075000 D
0C10  4809 0000 C000 C0F0          BMI                                                 04076000 D
0C11  4809 0C40 1000 00FC          B = A3                                              04077000 D
```

```
0C12  4812 0000 0030 00F0                 SAVE
0C13  4809 E15E 0C00 0CF0                 A3 GEQ LIT            % CHECK FOR SHIFT > 16          04079C00 D
0C14  7C29 E15E 1000 00F0                 IF TRUE THEN A3 + NOT LIT + 1 = A3; JUMP             04080C00 D
0C15  4809 E000 0B30 00FC                 A3 = B                                               04081000 D
0C16  4809 2C5E 0030 80F0                 LIT - B = SAR                                        04082C00 D
0C17  4809 C0C1 8B3C 00F0                 A2 C = B                                             04083C00 D
0C18  4809 0C41 0B30 00F0                 B L = B                                              04084C00 D
0C19  000C C000 C000 0020                 16 = SAR                                             04085C00 D
0C1A  4809 0C40 8B30 00FC                 D R = MIR, B                                         04086C00 D
0C1B  2F50 0000 0000 0060                 EOUTPUT - 1 = CPCR                                   04087C00 D
0C1C  2000 00C0 009C 0060                 SETCCA - 1 = CPCR     % SET THE CONDITION BITS       04088000 D
0C1D  56E0 00C0 0C00 0040                 OPCODE - 1 = MPCR                                    04089C00 D
                                                                %                              04090C00 D
                                                                %                              04091C0C D
                                  OP153:                        % RX TYPE STORE AND INDEX BY 1  04092C00 D
                                                                % (R(A)) INTO Y, (R(M)) + 1 -> R(M)  04093C00 D
0C1E  5700 00C0 0030 C060                 RXMFIELD - 1 = CPCR   % ADDR OF Y INTO B             04094C0C D
0C1F  4809 0C41 0020 00F0                 B L = DR1                                            04095C00 D
0C20  0000 00C0 0030 0030                 COMP B = SAR                                         04096C00 D
0C21  4809 E003 0B00 C0F0                 A3 = B                                               04097C00 D
0C22  22B0 00C0 0000 0060                 CONTENTSRA - 1 = CPCR  % (R(A)) INTO B               04098000 D
0C23  4809 0C40 003C 20F0                 B = MIR; ASR                                         04099000 D
0C24  4809 0F40 0C1C 00F0                 BMAR = BR2            % Y ADDR INTO ER2               04100C00 D
0C25  61EC 00C0 0C00 0060                 EMULOUT - 1 = CPCR                                   04101C00 D
0C26  2380 00C0 003C C060                 CONTENTSFM - 1 = CPCR  % (R(M)) INTO B               04102C00 D
0C27  4809 0C46 0030 00F0                 B + 1 = MIR                                          04103C0C D
0C28  2F50 0000 0030 0060                 EOUTPUT - 1 = CPCR                                   04104C0C D
0C29  66E0 C0C3 0000 0040                 OPCODE - 1 = MPCR                                    04105000 D
                                                                %                              04106C0C D
                                                                %                              04107C00 D
                                  OPCODE16:                                                    04108C00 D
0C2A  5F90 00C0 0030 C060                 XFCODE - 1 = CPCR     % "F" INTO A2                  04109C00 D
0C2B  4809 C640 0C40 C0F0                 A2 + AMPCR = AMPCR                                   04110C0C D
0C2C  15F0 0000 0C30 00C0                 OP16F - 1 = AMPCR                                    04111C0C D
0C2D  4809 00C0 0000 C0F0                 STEP                                                 04112C00 D
0C2E  4824 0000 0000 00FC                 EXEC                                                 04113C00 D
                                                                %                              04114C00 D
0C2F  160C 00C3 0030 0040         OP16F:  OP160 - 1 = MPCR                                     04115C00 D
0C30  1610 00C0 009C 0040                 OP161 - 1 = MPCR                                     04116C00 D
0C31  1620 00C3 0C0C 0040                 OP162 - 1 = MPCR                                     04117C00 D
0C32  163C 0000 0030 C040                 OP163 - 1 = MPCR                                     04118C00 D
                                                                %                              04119000 D
                                  OP160:                        % RR TYPE ALGEBRAIC LEFT DOUBLE SHIFT  04120C00 D
                                                                % SHIFT (R(A),R(A+1)) LEFT (R(M)) (0-5)  04121C00 D
                                                                % ZERO FILL AND SET CC, SET OVERFLOW  04122000 D
0C33  4809 0C40 9B0C 00F0                 B R = B                                              04123C00 D
0C34  80FC 00C0 0000 0080                 Q = SAR; 15 = LIT                                    04124000 D
0C35  4809 2C56 0B30 00F0                 LIT AND B = B         % ISOLATE THE "A" FIELD         04125000 D
0C36  51CC 0000 0C30 0060                 REGSTACK - 1 = CPCR   % ADDR OF R(A) INTO MAR2       04126C0C D
0C37  2F3C 00C3 C000 006C                 EINPUT - 1 = CPCR     % (R(A)) INTO B                04127C00 D
0C38  4809 0C41 203C 00F0                 B L = A2              % TEMP STORAGE OF (R(A))        04128000 D
0C39  001C 09C0 0030 C0AC                 COMP 16 = SAR; 1 = LIT                               04129C00 D
0C3A  4809 0F41 0020 00F0                 BMAR L = BR1          % TEMP STORAGE OF R(A)          04130C0C D
0C3B  0000 00C0 0000 0030                 COMP 8 = SAR                                         04131C00 D
0C3C  4809 2F5C 001C 00F0                 LIT OR BMAR = MAR2    % R(A+1)                       04132C00 D
0C3D  2F3C 00C0 0000 CC6C                 EINPUT - 1 = CPCR     % (R(A+1)) INTO B              04133C0C D
0C3E  4809 CC5C 203C 00FC                 A2 OR B = A2          % A2 = (R(A))/(R(A+1))          04134C00 D
0C3F  4809 E156 0B3C 00F0                 A3 AND LIT = B        % ISOLATE "M" FIELD             04135C0C D
0C40  00FC 00C0 0030 00E0                 15 = LIT                                             04136C00 D
0C41  51C0 0000 009C 0060                 REGSTACK - 1 = CPCR                                  04137C00 D
```

```
0C43  4809 0C5E 0030 80F0              -8 = SAR                                                  04139000 D
0C44  4809 C000 9070 C0F0              A2 R = A3               % FOR OVERFLOW TEST               04140000 D
0C45  4809 EC02 0000 C0F0              NOT A3                                                    04141000 D
0C46  63C9 0FC0 0C30 00F0              IF NOT ABT THEN SET LC1                                   04142000 D
0C47  3020 C000 0000 0060              OVBIT - 1 = CPCR         % SET OVERFLOW BIT               04143000 D
0C48  4809 0C5E 0030 80F0              -8 = SAR                                                  04144000 D
0C49  49C9 C001 2800 00F0              A2 L = A2, B; SET LC1                                     04145000 D
0C4A  2000 00C0 0000 006C              SETCCA - 1 = CPCR                                         04146000 D
0C4B  4809 C0C0 0C00 20F0              ASR                      % REFERENCE BR1                  04147000 D
0C4C  4809 0F43 801C 00F0              BMAR R = MAR2                                             04148000 D
0C4D  000C 00C0 0000 0010              B = SAR                                                   04149000 D
0C4E  48C9 C0C0 8030 00F0              A2 R = MIR                                                04150000 D
0C4F  00C0 0000 0C00 C020              16 = SAR                                                  04151000 D
0C50  2F50 00C0 0C0C 0060              EOUTPUT - 1 = CPCR       % WRITE OUT SHIFTED (R(A))       04152000 D
0C51  0010 00C0 00C0 00A0              COMP 16 = SAR; 1 = LIT                                    04153000 D
0C52  4809 C001 2000 00F0              A2 L = A2                                                 04154000 D
0C53  4809 C0C0 8030 C0F0              A2 R = MIR                                                04155000 D
0C54  48C9 2F5C 001C 00F0              LIT OR BMAR = MAR2       % R(A+1)                         04156000 D
0C55  2F50 00C0 0030 0060              EOUTPUT - 1 = CPCR                                        04157000 D
0C56  56EC 0003 00C0 C040              OPCODE - 1 = MPCR                                         04158000 D
                                       %                                                         04159000 D
                                       %                                                         04160000 D
                                OP161: %  RI TYPE 2 STORE AND INDEX BY 2                         04161000 D
                                       %  (R(A),R(A+1)) INTO Y**, Y**+1                         04162000 D
                                       %  (R(M)) + 2 INTO R(M)                                  04163000 D
0C57  238C 0000 C070 0060              CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                    04164000 D
0C58  4809 0C41 002C 00F0              B L = BR1              % Y* INTO ER1                      04165000 D
0C59  002C C0C0 0000 C0B0              COMP 8 = SAR; 2 = LIT                                     04166000 D
0C5A  4809 2C40 003C 00F0              LIT + B = MIR                                             04167000 D
0C5B  4809 0F41 0C1C 00F0              BMAR L = BR2                                              04168000 D
0C5C  51E0 0000 0000 C06C              EMULOUT - 1 = CPCR     % (R(M)) + 2 INTO R(M)             04169000 D
0C5D  48C9 EC00 8800 00F0              A3 R = B                                                  04170000 D
0C5E  80FC 0000 0070 D080              15 = LIT; 4 = SAR                                         04171000 D
0C5F  48C9 2C56 0B30 00F0              LIT AND B = B          % ISOLATE "A"                      04172000 D
0C60  51C0 0000 0C00 0060              REGSTACK - 1 = CPCR      \                               04173000 D
0C61  4809 E0C0 C830 C0F0              A3 = B                                                    04174000 D
0C62  22BC 0C00 000C C060              CONTENTSRA - 1 = CPCR % (R(A)) INTO E                     04175000 D
0C63  48C9 0C40 003C C0F0              B = MIR                                                   04176000 D
0C64  4809 00C0 003C 20F0              ASR                    % REFERENCE BR1                    04177000 D
0C65  4809 0F43 0010 00F0              BMAR = BR2                                                04178000 D
0C66  61EC 00C3 0070 D06C              EMULOUT - 1 = CPCR     % (R(A)) INTO Y*                   04179000 D
0C67  48C9 0F47 0020 00F0              BMAR + 1 L = BR1       % ADDR OF Y* + 1                   04180000 D
0C68  000C 00C0 0000 C030              COMP 8 = SAR                                              04181000 D
0C69  4809 E0C3 801C C0F0              A3 R = MAR2            % ADDR OF R(A)                     04182000 D
0C6A  0010 0000 0000 C0AC              16 = SAR; 1 = LIT                                         04183000 D
0C6B  4809 2F5C 001C C0F0              LIT OR BMAR = MAR2     % R(A+1)                          04184000 D
0C6C  2F3C C0C3 0C30 C060              EINPUT - 1 = CPCR      % (R(A+1)) INTO B                 04185000 D
0C6D  4809 0C40 003C C0F0              B = MIR                                                   04186000 D
0C6E  4809 00C0 0C30 20F0              ASR                    % REFERENCE BR1                    04187000 D
0C6F  4809 0F43 0010 00F0              BMAR = BR2                                                04188000 D
0C70  61EC 0003 0070 C060              EMULOUT - 1 = CPCR     % (R(A+1)) INTO Y* +1              04189000 D
0C71  66E0 0000 CC30 0040              OPCODE - 1 = MPCR                                         04190000 D
                                       %                                                         04191000 D
                                       %                                                         04192000 D
                                OP162: %  RK TYPE ALG LEFT DOUBLE SHIFT                          04193000 D
                                       %  SHIFT (R(A),R(A+1) LEFT Y (0-5) PLACES                04194000 D
                                       %  ZERO FILL, SET CC AND SET OVERFLOW BIT                04195000 D
0C72  4809 2C55 0B30 00F0              LIT AND B = B          % ISOLATE "M"                      04196000 D
0C73  00FC 0000 0000 00EC              15 = LIT                                                  04197000 D
```

```
0C74  4809 0C52 0000 00F0          0 EQL B                                          04198C00 D
0C75  6819 0C00 C030 00F0          IF TRUE THEN SKIP                                04199C00 D
0C76  2386 0000 0C00 0060          CONTENTSPM - 1 = CPCR  % (R(M)) INTO B            04200000 C
0C77  4809 0C41 0B3C 00F0          0 L = B                                          04201C00 D
0C78  0000 0F00 C00C 0020          16 = SAR                                         04202C00 D
0C79  4809 EC5C 1C30 00F0          A3 OR B = A3          % A3 = (R(M))/INSTRUCTION   04203C00 D
0C7A  6330 0000 0000 C060          IFETCH - 1 = CPCR     % "Y" INTO B               04204C00 D
0C7B  4809 E0C0 AC30 00F0          A3 R = A2                                        04205C00 D
0C7C  0000 00C0 003C 0020          16 = SAR                                         04206C00 D
0C7D  4809 CC40 C030 00F0          A2 + B = MIR          % Y INTO MIR               04207C00 D
0C7E  4809 E0C0 903C 00F0          A3 R = A3                                        04208C00 D
0C7F  30F0 0303 0CC0 0080          4 = SAR; 15 = LIT                                04209C00 D
0C80  4809 E155 0B70 00F0          A3 AND LIT = B        % ISOLATE "A"              04210C0C D
0C81  51C0 00C0 0000 C060          REGSTACK - 1 = CPCR                              04211C00 D
0C82  4809 0F41 0020 C0F0          BMAR L = BR1          % TEMP STORAGE OF R(A)      04212C00 D
0C83  000C 00C0 0090 C030          COMP 8 = SAR                                     04213C00 D
0C84  2F3C 0000 003C C060          EINPUT - 1 = CPCR     % (R(A)) INTO B            04214C00 D
0C85  4809 0C41 103C 00F0          B L = A3                                         04215C00 D
0C86  001C 000D 003C 00A0          COMP 16 = SAR; 1 = LIT                           04216C0C D
0C87  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2    % R(A+1)                   04217000 D
0C88  2F3C C000 0030 C060          EINPUT - 1 = CPCR                                04218C00 D
0C89  4809 EC5C 2000 00F0          A3 OR B = A2, BMI     % A2 = (R(A))/(R(A+1)), B = Y 04219C00 D
0C8A  4809 0C5E 0000 80F0          -B = SAR              % SET UP SHIFT AMOUNT       04220000 D
0C8B  4809 C000 9000 00F0          A2 R = A3             % BITS LOST IN LEFT SHIFT   04221000 D
0C8C  4809 E002 0000 00F0          NOT A3                                           04222C00 D
0C8D  63C9 C0C0 0C00 00F0          IF NOT ABT THEN SET LC1                          04223000 D
0C8E  5020 0C03 000C C06C          OVBIT - 1 = CPCR      % SET THE OVERFLOW BIT     04224C00 D
0C8F  4809 0C5E 0000 80F0          -B = SAR              % SHIFT AMOUNT             04225C00 D
0C90  49C9 C0C1 1B7C C0F0          A2 L = A3,B; SET LC1  % PERFORM SHIFT            04226C0C D
0C91  20DC 000D 0000 C06C          SETCCA - 1 = CPCR     % SET THE CONDITION BITS    04227C00 C
0C92  4809 E0C1 2000 C0F0          A3 L = A2             % (R(A+1)) INTO MS WORD OF A2 04228C00 D
0C93  0000 0000 0C00 0020          COMP 16 = SAR                                    04229C00 D
0C94  4809 C0C0 8030 00F0          A2 R = MIR            % (R(A+1)) INTO MIR         04230C00 D
0C95  2F50 0000 003C C06C          EOUTPUT - 1 = CPCR                               04231C00 D
0C96  4809 E0C0 8030 00F0          A3 R = MIR                                       04232C0C D
0C97  4809 0000 0000 2CF0          ASR                   % REFERENCE BR1            04233C00 D
0C98  4809 CF4D 801C C0F0          BMAR R = MAR2                                    04234C00 D
0C99  0C00 0000 007C 0010          8 = SAR                                          04235C00 D
0C9A  2F5C 0003 0070 0060          EOUTPUT - 1 = CPCR                               04236C00 D
0C9B  66EC 0000 C000 0040          OPCODE - 1 = MPCR                                04237C00 D
                                                 %                                 04238C00 D
                                                 %                                 04239C00 D
                   OP163:                        % RX TYPE STORE AND INDEX EY 2     04240C000 C
                                                 % (R(A),R(A+1)) INTO Y, Y+1       04241C00 D
                                                 % (R(M)) + 2 INTO R(M)            04242C0C D
0C9C  570C 0000 0C00 006C          RXMFIELD - 1 = CPCR   % Y INTO B                 04243C00 D
0C9D  4809 0C41 0C20 C0F0          B L = BR1             % TEMP STORAGE OF Y INTO BR1 04244C00 D
0C9E  00C0 00C0 0C70 0030          COMP 8 = SAR                                     04245C00 D
0C9F  238C 000D 003C 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B           04246C00 D
0CA0  4809 2C40 0030 C0FC          LIT + B = MIR                                    04247C00 D
0CA1  002C C0C0 0C3C 00E0          2 = LIT                                          04248C00 D
0CA2  2F50 00C3 0C00 C06C          EOUTPUT - 1 = CPCR    % (R(M)) + 2 INTO R(M)     04249C00 D
0CA3  4809 E0C3 9000 00F0          A3 R = A3                                        04250C00 D
0CA4  80FC 00C3 C000 008C          4 = SAR; 15 = LIT                                04251C00 D
0CA5  4809 E156 0B0C 00F0          A3 AND LIT = B        % ISOLATE "A"              04252C00 D
0CA6  51CC 00C3 C030 0060          REGSTACK - 1 = CPCR   % ADDR OF R(A) INTO MAR2   04253CC0 D
0CA7  2F3C 0003 0C00 0060          EINPUT - 1 = CPCR     % (R(A)) INTO B            04254C00 D
0CA8  4809 0C40 0030 00F0          B = MIR                                          04255C00 D
0CA9  4809 2F5C 1C30 C0F0          LIT OR BMAR = A3      % R(A+1)                   04256C00 D
0CAA  001C 00C3 0000 00E0          1 = LIT                                          04257CC0 C
```

```
0CAB  4809 00C0 C000 20FC              ASR                        % REFERENCE BR1          04258C00 D
0CAC  4809 0F43 001C 00F0              BMAR = BR2                 % ADDR OF Y INTO BR2     04259C00 D
0CAD  61E0 0000 0000 0060              EMULOUT - 1 = CPCR         % (R(A)) INTO Y          0426CC0C D
0CAE  4809 0F47 0020 00F0              BMAR + 1 L = BR1           % ADDR OF Y + 1          04261C00 D
0CAF  0C00 00C3 0030 0C30              COMP 8 = SAR                                       04262C0C D
0CD0  4809 E003 C01C 00F0              A3 = MAR2                                          04263C0C D
0C91  2F3C C0C3 0000 006C              EINPUT - 1 = CPCR          % (R(A+1)) INTO B        04264CC0 D
0CB2  4809 0C4C 0030 00F0              B = MIR                                            04265C00 D
0CB3  4809 C0C0 0000 20FC              ASR                        % REFERENCE BR1          04266C00 D
0CB4  4809 0F43 001C 00F0              BMAR = BR2                                         04267C00 D
0CB5  61E0 C000 0000 006C              EMULOUT - 1 = CPCR         % (R(A+1)) INTO Y+1      04268C00 D
0CB6  66EC 00C0 0C3C C040              OPCODE - 1 = MPCR                                  04269C00 D
                                                                  %                        0427C0C0 D
                                                                  %                        04271C00 D
0CB7  5F9C 0000 000C 0060      OPCODE17:  XFCODE - 1 = CPCR       % "F" INTO A2            04272000 D
0CB8  4809 C640 0C90 C0F0              A2 + AMPCR = AMPCR                                 04273C00 D
0CB9  1640 00C0 0000 C0C0              OP17F - 1 = AMPCR                                  04274C00 D
0CBA  4809 00C0 0000 00F0              STEP                                               04275C00 D
0CBD  4824 00C0 0C0C 00FC              EXEC                                               04276C0C D
                                                                  %                        0427CC00 D
                                                                  %                        04278C00 D
0CBC  1650 00C0 000C 004C      OP17F:  OP17C - 1 = MPCR                                   04279C00 D
0CBD  166C 0000 0030 0040              OP171 - 1 = MPCR                                   0428CC00 D
0CBE  167C 00C0 C03C C040              OP172 - 1 = MPCR                                   04281C00 D
0CDF  168C 00C0 0C3C C040              OP173 - 1 = MPCR                                   04282C00 D
                                                                  %                        04283C00 D
                                                                  %                        04284000 D
                                       OP170:                     % RR TYPE CIRCULAR DOUBLE LEFT SHIFT  04285C00 D
                                                                  % SHIFT (R(A),R(A+1)) LEFT CIRCULARLY 04286C00 D
                                                                  % (R(M)) (0-5) PLACES AND SET CC      04287C0C D
0CC0  2380 C003 0C00 C060              CONTENTSRM - 1 = CPCR      % (R(M)) INTO B          04288CC0 D
0CC1  4809 CC40 20FC 00F0              B = A2                                             04289C00 D
0CC2  48C9 E0C0 8B0C C0F0              A3 R = B                                           0429CC0C D
0CC3  80F0 0C00 0000 0080              4 = SAR; 15 = LIT                                  04291C00 D
0CC4  4809 2C56 0B9C 00F0              LIT AND B = B                                      04292C00 D
0CC5  51CC 00C0 CC0C 0060              REGSTACK - 1 = CPCR                                04293C00 D
0CC6  48C9 0F41 0020 00F0              BMAR L = BR1               % TEMP STORAGE OF R(A)   0429ACAC D
0CC7  001C C000 0030 00B0              COMP 8 = SAR; 1 = LIT                              04295C00 D
0CC8  4809 2F5C 001C 00F0              LIT OR BMAR = MAR2         % R(A+1)                 04296C00 D
0CC9  2F3C 00C0 0030 C060              EINPUT - 1 = CPCR          % (R(A)) INTO B          04297C0C D
0CCA  4809 0F46 001C C0FC              BMAR + 1 = MAR2                                    04298CC0 D
0CCB  4809 CC41 1C9C 00FC              B L = A3                   % A3 HAS (R(A)) IN MS WORD  04299C00 D
0CCC  000C 00C0 0030 0020              COMP 16 = SAR                                      0430CC00 D
0CCD  2F3C 00C0 0C30 0060              EINPUT - 1 = CPCR          % (R(A+1)) INTO B        04301C00 D
0CCE  4809 EC5C 1030 00F0              A3 OR B = A3               % A3 = (R(A))/(R(A+1))   0430C2C0 D
0CCF  4809 C0C2 2000 C0FC              NOT A2 = A2                                        04303CC0 D
0CD0  4809 CCC0 C000 80F0              A2 + 1 = SAR               % COMP OF (R(M)) (C-5)   0430C4C0C D
0CD1  49C9 E0C1 9B9C 00F0              A3 C = A3; B; SET LC1      % PERFORM SHIFT          04305CC0 D
0CD2  2000 0000 0030 0060              SETCCA - 1 = CPCR          % SET THE CONDITION BITS  0430C6C0 D
0CD3  48C9 E0C1 0B3C 00F0              A3 L = B                   % (R(A+1)) INTO B        0430C7C00 D
0CD4  000C 00C0 0C3C 0020              COMP 16 = SAR                                      0430C8C00 D
0CD5  4809 0C40 8030 C0F0              B R = MIR                  % (R(A+1)) INTO MIR      04309C00 D
0CD6  2F50 00C3 CC3C C060              EOUTPUT - 1 = CPCR         % WRITE SHIFTED (R(A+1))  0430AC0C D
0CD7  48C9 00C3 000C 20F0              ASR                        % REFERENCE BR1          0430BC00 D
0CD8  4809 0F43 8C1C C0F0              BMAR R = MAR2                                      0430C3C0 D
0CD9  000C 00C0 C030 0010              B = SAR                                            04313C00 D
0CDA  4809 E0C0 8030 00F0              A3 R = MIR                                         0430C4C0C D
0CDB  000C C0C0 0030 CC2C              16 = SAR                                           04315C00 D
0CDC  2F5C 00C0 0C3C C06C              EOUTPUT - 1 = CPCR                                 0430C6C0 D
0CDD  66E0 0000 C00C 004C              OPCODE - 1 = MPCR                                  04317C00 D
```

```
                                                              *                                        04319000 D
                                   OP171:                     * RI TYPE 1 STORE ZERO, 0 INTO Y*        04320000 D
0CDE  2380 0000 0000 0060             CONTENTSRM - 1 = CPCR   * (R(M)) INTO B                          04321000 D
0CDF  4809 0C41 0010 00F0             B L = BR2                                                        04322000 D
0CE0  0000 0000 0C00 0030             COMP 8 = SAR                                                     04323000 D
0CE1  4809 00C0 0030 C0F0             0 = MIR                                                          04324000 D
0CE2  61E0 0000 0000 0060             EMULOUT - 1 = CPCR                                               04325000 D
0CE3  66E0 0000 0000 0040             OPCODE - 1 = MPCR                                                04326000 D
                                                              *                                        04327000 D
                                                              *                                        04328000 D
                                   OP172:                     * RK TYPE CIRCULAR LEFT DOUBLE SHIFT     04329000 D
                                                              * SHIFT (R(A),R(A+1)) LEFT CIR. Y (0-5)  04330000 D
                                                              * AND SET CC                             04331000 D
0CE4  4809 2C56 0B00 00F0             LIT AND B = B                                                    04332000 D
0CE5  00F0 C000 0000 0CE0             15 = LIT                                                         04333000 D
0CE6  4809 0C52 C000 00F0             C EQL B                                                          04334000 D
0CE7  6819 0000 0000 00F0             IF TRUE THEN SKIP                                                04335000 D
0CE8  238C 0000 0000 0060             CONTENTSRM - 1 = CPCR   * (R(M)) INTO B                          04336000 D
0CE9  4809 0C41 0B00 00F0             B L = B                                                          04337000 D
0CEA  0000 0000 0000 C020             COMP 16 = SAR                                                    04338000 D
0CEB  4809 EC5C 1000 00F0             A3 OR B = A3            * A3 = (R(M))/INSTRUCTION                04339000 D
0CEC  6330 0000 0000 0060             IFETCH - 1 = CPCR       * "Y" INTO B                             04340000 D
0CED  4809 ECC0 A000 00F0             A3 R = A2               * (R(M)) INTO A2                         04341000 D
0CEE  000C 0000 0000 0020             16 = SAR                                                         04342000 D
0CEF  4809 CC40 2000 00FC             A2 + B = A2             * Y INTO A2                              04343000 D
0CF0  4809 ECC0 9000 C0F0             A3 R = A3                                                        04344000 D
0CF1  80F0 00C0 0000 0080             4 = SAR; 15 = LIT                                                04345000 D
0CF2  4809 E155 0B00 C0F0             A3 AND LIT = B          * ISOLATE "A"                            04346000 D
0CF3  51C0 0000 0000 0060             REGSTACK - 1 = CPCR                                              04347000 D
0CF4  4809 0F41 0C20 00F0             BMAR L = BR1            * TEMP STORAGE OF R(A)                   04348000 D
0CF5  001C 00C0 0000 C0B0             COMP 8 = SAR; 1 = LIT                                            04349000 D
0CF6  2F3C 0000 0000 0060             EINPUT - 1 = CPCR       * (R(A)) INTO B                          04350000 D
0CF7  4809 2F5C 001C 00F0             LIT OR BMAR = MAR2      * R(A+1)                                 04351000 D
0CF8  4809 0C41 1000 C0F0             P L = A3                * (R(A)) INTO FS WORD OF A3              04352000 D
0CF9  000C 0000 0000 0020             COMP 16 = SAR                                                    04353000 D
0CFA  2F30 C000 0000 C060             EINPUT - 1 = CPCR       * (R(A+1)) INTO B                        04354000 D
0CFB  4809 EC5C 1000 00F0             A3 OR B = A3            * A3 = (R(A))/(R(A+1))                   04355000 D
0CFC  4809 C002 200C 00F0             NOT A2 = A2                                                      04356000 D
0CFD  4809 C0C0 000C 80F0             A2 + 1 = SAR            * COMP OF Y (0-5)                        04357000 D
0CFE  49C9 EC01 9B00 C0F0             A3 C = A3,B; SET LC1    * PERFORM SHIFT                          04358000 D
0CFF  2000 00C0 0000 0060             SETCCA - 1 = CPCR       * SET THE CONDITION BITS                 04359000 D
0D00  4809 E001 2C00 C0F0             A3 L = A2                                                        04360000 D
0D01  000C 00C0 000C 0020             COMP 16 = SAR                                                    04361000 D
0D02  4809 C0C0 8080 C0F0             A2 R = MIR              * SHIFTED (R(A+1)) INTO MIF              04362000 D
0D03  2F5C 0000 0000 006C             EOUTPUT - 1 = CPCR                                               04363000 D
0D04  4809 0000 0000 20F0             ASR                     * REFERENCE BR1                          04364000 D
0D05  4809 0F40 8C1C C0F0             BMAR R = MAR2                                                    04365000 D
0D06  0000 0000 0000 C010             8 = SAR                                                          04366000 D
0D07  4809 E000 800C 00F0             A3 R = MIR                                                       04367000 D
0D08  000C 0000 0000 C020             16 = SAR                                                         04368000 D
0D09  2F50 0000 000C 006C             EOUTPUT - 1 = CPCR                                               04369000 D
0D0A  66E0 0000 0000 0040             OPCODE - 1 = MPCR                                                04370000 D
                                                              *                                        04371000 D
                                                              *                                        04372000 D
                                   OP173:                     * RX TYPE STORE ZERO, C INTO Y           04373000 D
0D0B  57C0 0000 0000 0060             RXMFIELD - 1 = CPCR     * ADDR OF Y INTO D                       04374000 D
0D0C  4809 0C41 0010 00F0             B L = BR2                                                        04375000 D
0D0D  0000 0000 000C C030             COMP 8 = SAR                                                     04376000 D
0D0E  4809 0CC0 0000 00F0             C = MIR                                                          04377000 D
```

```
0010  66E0 0000 0030 0040          OPCODE - 1 = MPCR                                  04379C00 D
                                                %                                    04380000 D
                                                %                                    04381000 D
0011  5F9C 000C 0000 0060    OPCODE20:  XFCODE - 1 = CPCR    % "F" INTO A2           04382000 D
0012  4809 C640 0C40 00F0          A2 + AMPCR = AMPCR                                 04383000 D
0013  169C 0000 0030 00C0          OP20F - 1 = AMPCR                                  04384000 D
0014  4809 000C 0C00 00F0          STEP                                               04385000 D
0015  4824 0003 0030 00F0          EXEC                                               04386000 D
                                                %                                    04387000 D
0016  16A0 00C0 0000 0040    OP20F:  OP200 - 1 = MPCR                                 04388000 D
0017  16BC 0000 0030 0040          OP201 - 1 = MPCR                                   04389000 D
0018  16CC 00C0 0000 0040          OP202 - 1 = MPCR                                   0439C000 D
0019  16DC 0000 003C 0040          OP203 - 1 = MPCR                                   04391000 D
                                                %                                    04392000 D
                                                %                                    04393000 D
                                OP200:                          % RR TYPE SUBTRACT REGISTER  04394000 D
                                                % (R(A)) - (R(M)) INTO R(A)           04395000 D
                                                % SET CC, CARRY AND OVERFLOW         04396000 D
001A  22B0 00F0 0C00 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B             04397000 D
001B  2809 0C41 203C 00F0          B L = A2; IF LC1       % (R(A)) INTO MS WORD OF A2  04398000 D
001C  00F0 0C00 0000 00A0          COMP 16 = SAR; 15 = LIT                            04399000 D
001D  4809 E155 0B30 00F0          A3 AND LIT = B                                     04400000 D
001E  51C0 00F0 0030 C06C          REGSTACK - 1 = CPCR                                04401000 D
001F  2F30 0000 C000 0060          EINPUT - 1 = CPCR      % (R(M)) INTO E             04402000 D
0020  4849 0C41 0B00 00F0          B L = B; SET LC2                                   04403000 D
0021  4809 CC5E 003C 00F0          A2 - B = MIR           % (R(A)) - (R(M)) INTO MIR  04404000 D
0022  7BC9 0000 0030 00F0          IF ADV THEN SET LC1                                04405000 D
0023  1E70 0000 000C 0060          CARRY - 1 = CPCR       % SET THE CARRY BIT         04406000 D
0024  9F10 00F3 0030 0060          CHECKOV - 1 = CPCR     % SET THE OVERFLOW BIT      04407000 D
0025  4809 0000 0D3C 00F0          BMI                                                04408000 D
0026  4809 0C40 8B30 00F0          B R = B,MIR                                        04409000 D
0027  00F0 0000 000C 0020          16 = SAR                                           04410000 D
0028  4809 E003 801C 00F0          A3 R = MAR2            % ADDR OF R(A)              04411000 D
0029  2F5C 0003 0000 0060          EOUTPUT - 1 = CPCR                                 04412000 D
002A  2CD0 00C0 00B0 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS    04413000 D
002B  66E0 0003 0000 0040          OPCODE - 1 = MPCR                                  04414000 D
                                                %                                    04415000 D
                                                %                                    04416000 D
                                OP201:                          % RI TYPE 2 SUBTRACT (INDIRECT)  04417000 D
                                                % (R(A)) - (Y*) INTO R(A),           04418000 D
                                                % SET CC, CARRY, AND OV BITS         04419000 D
002C  2380 0C00 003C C060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B             04420000 D
002D  4809 0C41 0020 C0FC          B L = BR1              % Y* INTO BR1                04421000 D
002E  0600 00F0 0030 C03C          COMP B = SAR                                       04422000 D
002F  4809 E000 0B00 00F0          A3 = B                                             04423000 D
0030  22BC 00C3 003C 006C          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B             04424000 D
0031  4809 0C41 203C C0F0          B L = A2               % (R(A)) INTO MS WORD OF A2  04425000 D
0032  0000 00C0 003C 0020          COMP 16 = SAR                                      04426000 D
0033  4809 000C 0030 20F0          ASR                    % DEF BR1                   04427000 D
0034  4809 0F43 0C10 C0F0          BMAR = BR2                                         04428000 D
0035  60E0 0003 0030 0060          EMULIN - 1 = CPCR      % (Y*) INTO B               04429000 D
0036  4809 0C41 0B3C C0F0          B L = B                % (Y*) INTO MS WORD OF B     0443C000 D
0037  0000 0000 0030 C02C          COMP 16 = SAR                                      04431000 D
0038  4849 CC5E 003C 00FC          A2 - B = MIR; SET LC2 % (R(A)) - (Y*) INTO MIR     04432000 D
0039  7BC9 00C0 003C 00F0          IF ADV THEN SET LC1                                04433000 D
003A  1E70 0000 0C3C 0060          CARRY - 1 = CPCR                                   04434000 D
003B  9F10 00C0 0000 0060          CHECKOV - 1 = CPCR                                 04435000 D
003C  4809 00C0 0D3C 00F0          BMI                                                04436000 D
003D  4809 0C43 8B3C 00F0          B R = B, MIR                                       04437000 D
```

```
0D3E  0000 0003 0030 0020          16 = SAR                                  04438C00 D
0D3F  4809 E0C3 801C 00F0          A3 R = MAR2          % ADDR OF R(A)       04439000 D
0D40  2F5C 0000 003C C060          EOUTPUT - 1 = CPCR   % WRITE OUT NEW R(A)  04440C00 D
0D41  2000 0000 0000 C06C          SETCCA - 1 = CPCR    % SET THE CONDITION BITS 04441800 D
0D42  66E0 0000 003C C040          OPCODE - 1 = MPCR                         04442000 D
                                   %                                         04443000 D
                                   %                                         04443C00 D
                              OP202:                    %TYPE RK SUBTRACT, (R(A)) -Y INTO R(A) 04444800 D
                                                        % SET CC, CARRY AND OV BITS 04445C00 D
0D43  4809 2C56 0B00 00FC          LIT AND B = B        % "M" INTO B          04446800 D
0D44  00F0 0000 003C 00EC          15 = LIT                                  04447400 D
0D45  4809 CC52 0C3C C0F0          C EQL B                                   04448000 D
0D46  6819 00C0 0000 00F0          IF TRUE THEN SKIP                         04448C00 D
0D47  2380 00C0 003C 006C          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B    04449800 D
0D48  4809 CC41 0B30 00F0          B L = B                                   04450400 D
0D49  C0D0 0000 0000 0020          COMP 16 = SAR                             04451000 D
0D4A  4809 EC5C 103C 00FC          A3 OR B = A3         % A3 = (R(M))/INSTRUCTION 04451C00 D
0D4B  633C 00C0 003C C06C          IFETCH - 1 = CPCR    % "Y" INTO B         04452800 D
0D4C  4809 E0D3 AC0C 00F0          A3 R = A2            % (R(M)) INTO A2      04453400 D
0D4D  00CC 0000 0000 0020          16 = SAR                                  04454000 D
0D4E  4809 CC41 003C 00F0          A2 + B L = MIR       % Y INTO MS WORD OF MIR 04454C00 D
0D4F  4809 E0C3 083C 00FC          A3 = B                                    04455800 D
0D50  2280 00C0 003C C060          CONTENTSRA - 1 = CPCR                     04456400 D
0D51  4809 0C41 2000 00F0          B L = A2, BMI        % (R(A)) INTO MS WORD OF A2 04457000 D
0D52  00CC 00C3 0030 002C          COMP 16 = SAR                             04457C00 D
0D53  4849 CC5E 0C30 00F0          A2 - B = MIR; SET LC2  % (R(A)) - (Y) INTO MIR 04458800 D
0D54  78C9 0000 003C C0F0          IF AOV THEN SET LC1                       04459400 D
0D55  1E7C 0000 003C 006C          CARRY - 1 = CPCR     % SET THE CARRY BIT  04460000 D
0D56  4F10 0000 0000 0060          CHECKOV - 1 = CPCR   % SET THE OV BIT      04460C00 D
0D57  4809 0003 003C 00F0          BMI                                       04461800 D
0D58  4809 0C40 883C 00F0          B R = D, MIR                              04462400 D
0D59  000C 00D3 C030 C020          16 = SAR                                  04463000 D
0D5A  2F50 CC03 0C30 0060          EOUTPUT - 1 = CPCR   % WRITE OUT NEW (R(A)) 04463C00 D
0D5B  2000 00C3 003C C060          SETCCA - 1 = CPCR    % SET THE CONDITION BITS 04464800 D
0D5C  66E0 0C00 003C 0040          OPCODE - 1 = MPCR                         04465400 D
                                   %                                         04466000 D
                                   %                                         04466C00 D
                              OP203:                    % TYPE RX SUBTRACT, (R(A))-(Y) -> R(A) 04467800 D
                                                        % SET CC, CARRY AND OV BITS 04468400 D
0D5D  5700 00C0 000C C060          RXMFIELD - 1 = CPCR  % ADDR OF Y INTO B    04469000 D
0D5E  4809 0C41 0C10 00FC          B L = BR2                                 04469C00 D
0D5F  00D0 00C3 0030 0030          COMP 8 = SAR                              04470800 D
0D60  60EC 00C3 0C0C C06C          EMULIN - 1 = CPCR    % (Y) INTO B         04471400 D
0D61  4809 0C41 0C30 00F0          D L = MIR            % (Y) INTO MS WORD OF MIR 04472000 D
0D62  000C 00C3 0C30 0020          COMP 16 = SAR                             04472C00 D
0D63  4809 EC00 0B3C 00F0          A3 = B                                    04473800 D
0D64  2280 0000 003C 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B    04474400 D
0D65  4809 0C41 2D3C C0F0          B L = A2, BMI        % (R(A)) INTO MS WORD OF A2 04475000 D
0D66  0000 0C03 003C C020          COMP 16 = SAR                             04475C00 D
0D67  4849 CC5E 0C3C 00F0          A2 - B = MIR; SET LC2  % (R(A)) - (Y) INTO MIR 04476800 D
0D68  78C9 0000 003C C0F0          IF AOV THEN SET LC1                       04477400 D
0D69  1E7C 00C3 0030 0060          CARRY - 1 = CPCR                          04478000 D
0D6A  4F10 00C0 0C30 C060          CHECKOV - 1 = CPCR                        04478C00 D
0D6B  4809 0003 003C C0FC          BMI                                       04479800 D
0D6C  4809 0C40 883C 00F0          B R = B, MIR                              04480400 D
0D6D  0000 0000 0000 C02C          16 = SAR                                  04481000 D
0D6E  2F5C 00C3 003C C060          EOUTPUT - 1 = CPCR   % WRITE NEW (R(A))    04481C00 D
0E6F  2000 00C0 0030 0060          SETCCA - 1 = CPCR    % SET THE CONDITION BITS 04482800 D
0D70  66E0 0003 C03C 0040          OPCODE - 1 = MPCR                         04483400 D
                                   %                                         04484000 D
```

```
                                                    %                                                   04498C00 D
0071  5F90 C000 0000 0060      OPCODE21:  XFCODE - 1 = CPCR     % "F" INTO A2                          04499C00 D
0072  4809 C640 0040 00FC                 A2 + AMPCR = AMFCR                                          04500C00 D
0073  16EC 0000 0030 00C0                 OP21F - 1 = AMPCR                                           04501C00 D
0074  4809 09C0 0000 00F0                 STEP                                                        04502C00 D
0075  4824 C000 0C0C C0F0                 EXEC                                                        04503C00 D
                                                    %                                                   04504C00 D
0076  16FC C000 0030 0040      OP21F:     OP210 - 1 = MPCR                                            04505C00 D
0077  1700 0000 0000 0040                 OP211 - 1 = MPCR                                            04506C00 D
0078  300C C000 0030 0040                 FAULT - 1 = MPCR                                            04507C00 D
0079  171C 00C3 0000 0040                 OP213 - 1 = MPCR                                            04508C00 D
                                                    %                                                   04509C00 D
                                                    %                                                   04510C00 D
                                OP210:              % TYPE RR SUBTRACT DOUBLE (REGISTER)             04511C00 D
                                                    % (R(A),R(A+1))-(R(M),R(M+1)) INTO               04512C00 D
                                                    % R(A),R(A+1), SET CC, OV, AND CARRY             04513C00 D
007A  4809 0C40 8800 00F0                 B R = B           % ISOLATE "A"                            04514C00 D
007B  30F0 00C0 0C30 C080                 4 = SAR; 15 = LIT                                         04515C00 D
007C  4809 2C55 0B00 00F0                 LIT AND B = B                                             04516C00 D
007D  51CC C0C0 0070 0060                 REGSTACK - 1 = CPCR                                       04517C00 D
007E  2F3C 00C3 0C00 006C                 EINPUT - 1 = CPCR       % (R(A)) INTO F                   04518C00 D
007F  4809 0C41 20DC C0F0                 B L = A2                % (R(A)) INTO MS WORD OF A2        04519C00 D
0080  0000 00C3 0C00 C020                 COMP 16 = SAR                                             04520C00 D
0081  4809 0F41 002C 00FC                 BMAR L = BR1            % TEMP STORAGE OF ADDR OF R(A)     04521C00 D
0082  001C 00C0 0C30 00B0                 COMP 8 = SAR; 1 = LIT                                     04522C00 D
0083  4809 2F55 0C1C 00F0                 LIT OR BMAR = MAR2      % R(A+1)                          04523C00 D
0084  2F3C 0000 C00C C060                 EINPUT - 1 = CPCR       % (R(A+1)) INTO B                 04524C00 D
0085  4809 CC55 2000 00F0                 A2 OR D = A2            % A2 = (R(A))/(R(A+1))             04525C00 D
0086  4809 E156 0E30 00F0                 A3 AND LIT = B          % ISOLATE "M"                      04526C00 D
0087  00F0 0000 0030 00EC                 15 = LIT                                                  04527C00 D
0088  51C0 0000 0C00 0C60                 REGSTACK - 1 = CPCR                                       04528C00 D
0089  2F3C 00C3 0000 0060                 EINPUT - 1 = CPCR       % (R(M)) INTO P                   04529C00 D
008A  4809 0C41 109C 00F0                 B L = A3                % (R(M)) INTO MS WORD OF A3        04530C00 D
008B  0010 00C3 0000 00A0                 COMP 16 = SAR; 1 = LIT                                    04531C00 D
008C  4809 2F5C 0F1C 00FC                 LIT OR BMAR = MAR2      % ADDR OF R(M+1)                  04532C00 D
008D  2F30 0000 0000 C060                 EINPUT - 1 = CPCR       % (R(M+1)) INTO B                 04533C00 D
008E  4809 EC5C 0B30 00F0                 A3 OR B = B             % D = (R(M))/(R(M+1))              04534C00 D
008F  4849 CC5E 0080 00F0                 A2 - D = MIR; SET LC2   % PERFORM SUBTRACTION             04535C00 D
0090  78C9 0C00 0030 00F0                 IF AOV THEN SET LC1                                       04536C00 D
0091  1E70 00C3 003C C060                 CARRY - 1 = CPCR                                          04537C00 D
0092  4F10 C000 0006 C060                 CHECKOV - 1 = CPCR                                        04538C00 D
0093  49C9 CFC0 0D00 C0F0                 BMI; SET LC1                                              04539C00 D
0094  200C 00C0 C00C 0060                 SETCCA - 1 = CPCR       % SET THE CONDITION BITS          04540C00 C
0095  4809 09C0 0D30 00F0                 BMI                                                       04541C00 D
0096  4809 0C40 8C90 C0F0                 B R = MIR               % NEW (R(A)) INTO MIR             04542C00 D
0097  0000 00C0 0C3C 002C                 16 = SAR                                                  04543C00 D
0098  4809 CC41 CD00 C0F0                 B L = B                                                   04544C00 C
0099  4809 0C40 8B0C 00F0                 B R = B                 % (R(A+1))                        04545C00 D
009A  4809 00C3 0030 20F0                 ASR                     % REFERENCE BR1                   04546C00 D
009B  4809 0F43 901C 00F0                 BMAR R = MAR2, A3       % ADDR OF R(A)                    04547C00 D
009C  000C 0FC0 0000 001C                 8 = SAR                                                   04548C00 D
009D  2F5C C0C3 0C30 006C                 EOUTPUT - 1 = CPCR      % WRITE NEW (R(A))                04549C00 D
009E  4809 EFDC 001C 00F0                 A3 OR 1 = MAR2          % ADDR OF R(A+1)                  04550C00 D
009F  4809 0C40 0C30 00F0                 B = MIR                 % (R(A+1))                        04551C00 D
00A0  2F50 0000 0000 C06C                 EOUTPUT - 1 = CPCR      % WRITE NEW (R(A+1))              04552C00 D
00A1  66EC C000 0030 C040                 OPCODE - 1 = MPCR                                         04553C00 D
                                                    %                                                   04554C00 C
                                                    %                                                   04555C00 D
                                OP211:              % RI TYPE 2 SUBTRACT DOUBLE                       04556C00 C
                                                    % (R(A),R(A+1))-(Y*,Y*+1) INTO                    04557C00 D
```

```
                                                                %R(A),R(A+1), SET CC, CARRY AND OV BITS   04558C00 D
0DA2  4809 0C40 8030 00F0                B R = B                                                          04559000 D
0DA3  80F0 00C0 0030 C080                4 = SAR, 15 = LIT                                               04560000 D
0DA4  4809 2C55 0830 00F0                LIT AND B = B          % ISOLATE "A"                            04561000 D
0DA5  51C0 00C0 003C C06C                REGSTACK - 1 = CPCR                                             04562000 D
0DA6  2F30 00C0 000C C06C                EINPUT - 1 = CPCR      % (R(A)) INTO E                          04563000 D
0DA7  4809 0C41 2030 00F0                B L = A2               % (R(A)) INTO MS WORD OF A2               04564000 D
0DA8  000C 00C0 003C C020                COMP 16 = SAR                                                   04565000 D
0DA9  4809 0F41 0C20 00F0                BMAR L = BR1           % SAVE ADDR OF R(A)                       04566000 D
0DAA  0010 00C0 0009 C080                COMP 8 = SAR, 1 = LIT                                           04567000 D
0DAB  4809 2F5C 0C1C 00F0                LIT OR BMAR = MAR2     % R(A+1)                                 04568000 D
0DAC  2F30 0000 0000 C06C                EINPUT - 1 = CPCR      % (R(A+1)) INTO B                        04569000 D
0DAD  4809 CC5C 2030 00F0                A2 OR B = A2           % A2 = (R(A))/(R(A+1))                   04570000 D
0DAE  4809 E156 0B00 00F0                A3 AND LIT = B         % ISOLATE "M"                            04571000 D
0DAF  00FC 0000 0030 C0E0                15 = LIT                                                        04572000 D
0DB0  51C0 0000 0030 006C                REGSTACK - 1 = CPCR     % (R(M)) INTO B                         04573000 D
0DB1  2F30 00C0 000C 0060                EINPUT - 1 = CPCR      % (R(M)) INTO G                          04574000 D
0DB2  4809 0C41 0C10 00F0                B L = BR2                                                       04575000 D
0DB3  000C 0C00 0030 003C                COMP 8 = SAR                                                    04576000 D
0DB4  50E0 00C0 003C 006C                EMULIN - 1 = CPCR      % (Y*) INTO B                            04577000 D
0DB5  4809 2F5D 001C 00F0                LIT OR BMAR L = ER2    % ADDR OF Y*+1                           04578000 D
0DB6  001C 00C0 000C 00B0                COMP 8 = SAR, 1 = LIT                                           04579000 D
0DB7  4809 0C41 1030 00F0                B L = A3               % (Y*) INTO MS WORD OF A3                 04580000 D
0DB8  000C 0030 0000 0020                COMP 16 = SAR                                                   04581000 D
0DB9  60E0 0000 003C 0060                EMULIN - 1 = CPCR      % (Y*+1) INTO F                          04582000 D
0DBA  4809 EC5C 0B0C 00F0                A3 OR B = B            % B = (Y*)/(Y*+1)                        04583000 D
0DBB  4849 CC5E 0C30 00F0                A2 - D = MIR, SET LC2  % PERFORM SUBTRACTION                    04584000 D
0DBC  78C9 0000 000C 00F0                IF AOV THEN SET LC1                                             04585000 D
0DBD  1E70 0000 0000 0C60                CARRY - 1 = CPCR                                                04586000 D
0DBE  4F1C 0C00 007C 0060                CHECKOV - 1 = CPCR     % SET THE OV BITS                        04587000 D
0DBF  49C9 0000 0D00 00F0                BMI, SET LC1                                                    04588000 D
0DC0  20D0 0000 0030 0060                SETCCA - 1 = CPCR      % SET THE CONDITION BITS                 04589000 D
0DC1  4809 0000 0030 00FC                BMI                                                             04590000 D
0DC2  4809 0C40 8C30 00F0                B R = MIR              % NEW (R(A))                             04591000 D
0DC3  0000 000C 0000 0020                16 = SAR                                                        04592000 D
0DC4  4809 0C41 0B30 00F0                B L = B                                                         04593000 D
0DC5  4809 0C40 8B30 00F0                B R = B                % NEW (R(A+1))                            04594000 D
0DC6  4809 000C 003C 20FC                ASR                    % REF BR1                                04595000 D
0DC7  4809 0F40 801C 00F0                BMAR R = MAR2          % ADDR OF R(A)                            04596000 D
0DC8  001C 00C0 0C00 C09C                B = SAR, 1 = LIT                                                04597000 D
0DC9  2F50 00C0 0000 0060                EOUTPUT - 1 = CPCR     % WRITE NEW (R(A))                        04598000 D
0DCA  4809 2F5C 001C 00FC                LIT OR BMAR = MAR2     % R(A+1)                                 04599000 D
0DCB  4809 0C40 0C30 00F0                B = MIR                % (R(A+1))                                04600000 D
0DCC  2F50 0000 0000 006C                EOUTPUT - 1 = CPCR     % WRITE NEW (R(A+1))                      04601000 D
0DCD  66E0 00C0 C07C 0040                OPCODE - 1 = MPCR                                               04602000 D
                                                                %                                        04603000 D
                                                                %                                        04604000 D
                                 OP213:                         % RX TYPE SUBTRACT DOUBLE                 04605000 D
                                                                % (R(A),R(A+1))-(Y,Y+1)=R(A),R(A+1)       04606000 D
                                                                % SET CC, OV AND CARRY BITS               04607000 D
0DCE  5700 00C0 0030 0060                RXMFIELD - 1 = CPCR    % Y ADDR INTO F                           04608000 D
0DCF  4809 0C41 0C10 00FC                B L = BR2                                                       04609000 D
0DD0  000C 00C0 0030 003C                COMP 8 = SAR                                                    04610000 D
0DD1  60E0 00C0 003C 0060                EMULIN - 1 = CPCR      % (Y) INTO B                             04611000 D
0DD2  4809 0C41 2030 00F0                B L = A2               % (Y) INTO MS WORD OF A2                  04612000 D
0DD3  000C 0000 0000 C02C                COMP 16 = SAR                                                   04613000 D
0DD4  4809 2F5D 0C1C 00F0                LIT OR BMAR L = ER2    % ADDR OF Y+1                            04614000 D
0DD5  001C 0000 0030 00BC                COMP 8 = SAR , 1 = LIT                                          04615000 D
0DD6  50E0 00C0 0C00 C060                EMULIN - 1 = CPCR      % (Y+1) INTO B                           04616000 D
0DD7  4809 CC5C 0090 00FC                A2 OR B = MIR          % MIR = (Y)/(Y+1)                        04617000 D
```

```
0D08  4809 E000 0800 C0FC          A3 = B
0D09  4609 0C40 8E30 C0F0          B R = B
0DDA  80F0 0000 0000 C080          4 = SAR; 15 = LIT
0DDB  4809 2C55 0B00 C0F0          LIT AND B = B
0DDC  51C0 0000 C0C0 006C          REGSTACK - 1 = CPCR
0DDD  2F30 0000 0000 C06C          EINPUT - 1 = CPCR      % (R(A)) INTO B
0DDE  4809 0F41 0C20 C0F0          BMAR L = BR1           % SAVE ADDR OF R(A)
0DDF  0000 0CC0 0000 C030          COMP 8 = SAR
0DE0  4809 0C41 2000 00FC          B L = A2               % (R(A)) INTO MS WORD OF A2
0DE1  0010 0000 0000 C0A0          COMP 16 = SAR; 1 = LIT
0DE2  4809 2F5C C01C C0FC          LIT OR BMAR = MAR2     % R(A+1)
0DE3  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B
0DE4  4809 CC5C 2D00 00F0          A2 OR B = A2, BMI      % A2 = (R(A))/(R(A+1))
0DE5  4849 CC5E 0050 00FC          A2 - B = MIR; SET LC2  % PERFORM SUBTRACTION
0DE6  78C9 C000 0000 00FC          IF AOV THEN SET LC1
0DE7  1E70 0CC0 0000 C060          CARRY - 1 = CPCR
0DE8  4F10 0000 000C C060          CHECKOV - 1 = CPCR
0DE9  49C9 0000 0DD0 00F0          BMI; SET LC1
0DEA  2000 0000 0000 006C          SETCCA - 1 = CPCR      % SET THE CONDITION BITS
0DEB  4809 0000 C000 C0F0          BMI
0DEC  4809 0C40 8090 00F0          B R = MIR              % NEW (R(A))
0DED  0000 C000 0000 C02C          16 = SAR
0DEE  4809 0C41 0B00 C0FC          B L = B
0DEF  4809 CC40 8600 00F0          B R = B
0DF0  4809 0000 0000 20F0          ASR                    % REF BR1
0DF1  4809 0F40 901C C0F0          BMAR R = MAR2  , A3
0DF2  C00C 0000 C000 0010          8 = SAR
0DF3  2F50 0CC0 0000 006C          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A))
0DF4  4809 E0DC C01C C0FC          A3 OR 1 = MAR2         % ADDR OF R(A+1)
0DF5  4809 0C40 0C3C 00F0          B = MIR
0DF6  2F50 0000 0000 C06C          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A+1))
0DF7  66EC 0C00 C00C C040          OPCODE - 1 = MPCR
                                                          %
                                                          %
0DF8  5F9C 0000 000C C060  OPCODE22:  XFCODE - 1 = CPCR   % "F" INTO A2
0DF9  4809 C640 0040 C0F0          A2 + AMPCR = AMPCR
0DFA  1720 0000 0000 00CC          OP22F - 1 = AMPCR
0DFB  4809 0000 0C00 C0F0          STEP
0DFC  4824 0000 C000 00F0          EXEC
                                                          %
0DFD  1730 C000 0000 C040  OP22F:  OP220 - 1 = MPCR
0DFE  1740 0000 0000 0040          OP221 - 1 = MPCR
0DFF  175C 0CC0 000C C040          OP222 - 1 = MPCR
0E00  176C 0000 0000 0040          OP223 - 1 = MPCR
                                                          %
                                                          %
                           OP220:                         % TYPE RR ADD, (R(A)) + (R(M)) -> R(A)
                                                          % SET CC, CARRY AND OV BITS
0E01  22B0 0000 0C00 0C60          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B
0E02  4809 0C41 2C0C C0F0          B L = A2
0E03  00FC C000 C0C0 00A0          COMP 16 = SAR; 15 = LIT
0E04  4809 E156 0B0C 00FC          A3 AND LIT = B
0E05  51C0 C0C0 0000 C060          REGSTACK - 1 = CPCR    % ADDR OF R(M) INTO MAR2
0E06  2F30 0000 C000 C060          EINPUT - 1 = CPCR      % (R(M)) INTO B
0E07  4809 0C41 0B00 C0FC          B L = B
0E08  000C 00C0 0000 0020          COMP 16 = SAR
0E09  4809 CC40 0C3C 0CF0          A2 + B = MIR           % PERFORM ADDITION
0E0A  78C9 0CC0 000C 00F0          IF AOV THEN SET LC1
0E0B  1E70 0000 C000 0060          CARRY - 1 = CPCR
0E0C  4F10 C000 C000 C060          CHECKOV - 1 = CPCR
```

```
0E0E  4809 0C40 8B30 00F0          B R = MIR, B                                    04679C00 D
0E0F  4809 0C40 8B30 C0FC          B R = MIR, B                                    04680C00 D
0E10  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR                              04681C0C D
0E11  20DC 0000 0000 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS 04682C00 D
0E12  66E0 0000 0000 0040          OPCODE - 1 = MPCP                               04683C0C D
                                                          %                        04684C00 D
                                                          %                        04685000 D
                                   OP221:                 % TYPE RI TYPE 2 ADD, (R(A))+(Y=) -> 04686CC0 D
                                                          % R(A), SET CC, CARRY, AND OV BITS 04687C00 D
0E13  22BC 0F00 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B          04688C0C D
0E14  4809 0C41 2C30 00FC          B L = A2               % (R(A)) INTO MS WORD OF A2 04689C00 D
0E15  00FC 0003 0030 00A0          COMP 16 = SAR; 15 = LIT                         04690C00 D
0E16  4809 E156 0B30 00F0          A3 AND LIT = B                                  04691000 D
0E17  51CC 00C0 C000 C06C          REGSTACK - 1 = CPCR                             04692000 D
0E18  2F30 0003 0000 0060          EINPUT - 1 = CPCR      % (R(M)) INTO B          04693C0C D
0E19  4809 0C41 0010 00FC          B L = BR2                                       04694000 D
0E1A  0000 0C00 0000 0030          COMP 8 = SAR                                    04695C0C D
0E1B  60E0 0000 0C30 006C          EMULIN - 1 = CPCR      % (Y=) INTO B            04696C00 D
0E1C  4809 0C41 0B0C 00F0          B L = B                % (Y=) INTO MS WORD OF B  04697C0C D
0E1D  0C0C 0003 0030 0C2C          COMP 16 = SAR                                   04698C0C D
0E1E  4809 CC40 0F90 00F0          A2 + B = MIR                                    04699C00 D
0E1F  7BC9 0000 C030 CCF0          IF AOV THEN SET LC1                             04700CC0 D
0E20  1E7C 00C3 0000 C060          CARRY - 1 = CPCR                                04701C00 D
0E21  4F1C 0CC0 0000 C060          CHECKOV - 1 = CPCR                              04702C00 D
0E22  4809 E0C3 8D1C C0FC          A3 R = MAR2, BMI       % ADDR OF R(A)           04703600 D
0E23  0000 0003 0030 0020          16 = SAR                                        04704C00 D
0E24  48C9 0C40 8B90 C0F0          B R = MIR, B                                    04705C00 D
0E25  2F50 0CC0 0000 0060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A))       04706C00 D
0E26  20DC 00C3 0030 0060          SETCCA - 1 = CPCR                               04707C00 D
0E27  66E0 0CC0 0C00 C040          OPCODE - 1 = MPCP                               04708C00 D
                                                          %                        04709C0C D
                                                          %                        04710C00 D
                                   OP222:                 % TYPE RK ADD, (R(A))+Y -> R(A) 04711C00 D
                                                          % SET CC, CARRY AND OV BITS 04712CC0 D
0E28  4809 2C56 0B30 00F0          LIT AND B = B                                   04713C00 D
0E29  00FC 00C0 0C00 C0E0          15 = LIT                                        04714C00 D
0E2A  4809 0C52 0030 00F0          C EQL B                                         04715C00 D
0E2B  5819 CCC0 0C30 0CFC          IF TRUE THEN SKIP                               04716C00 D
0E2C  238C 0CC0 0F0C 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B          04717C00 D
0E2D  4809 0C41 0B70 00FC          B L = B                % (R(M)) INTO MS WORD OF B 04718C00 D
0E2E  0000 00C3 0C00 C020          COMP 16 = SAR                                   04719C00 D
0E2F  4809 EC5C 1030 C0F0          A3 OR B = A3           % A3 = (R(M))/INSTRUCTION 04720C00 D
0E30  6330 0F00 CF30 0060          IFETCH - 1 = CPCR      % "Y" INTO B             04721CC0 D
0E31  4809 E0C0 A00C 00FC          A3 R = A2              % (R(M)) INTO A2          04722C00 D
0E32  0000 C0C0 0C0C 0020          16 = SAR                                        04723C00 D
0E33  4809 CC41 0C90 C0FC          A2 + B L = MIR         % Y INTO MIR              04724C00 D
0E34  4809 E0C3 0B30 C0F0          A3 = B                                          04725C00 D
0E35  228C 0CC0 0C0C C060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B          04726C00 D
0E36  4809 0C41 2D30 C0F0          B L = A2, BMI          % (R(A)) INTO MS WORD OF A2 04727C00 D
0E37  0000 00C3 0000 C02C          COMP 16 = SAR                                   04728C00 D
0E38  4809 CC40 0030 00F0          A2 + B = MIR           % PERFORM ADDITION        04729CC0 D
0E39  7BC9 C000 0C00 00F0          IF AOV THEN SET LC1                             04730CCC D
0E3A  1E7C 0CC0 0C30 C06C          CARRY - 1 = CPCR                                04731C00 D
0E3B  4F10 00C3 0030 C06C          CHECKOV - 1 = CPCR                              04732C00 D
0E3C  4809 0C03 0D00 00F0          BMI                                             04733C00 D
0E3D  4809 0C40 8B30 C0F0          B R = B, MIR                                    04734C00 D
0E3E  0000 0000 0C30 C02C          16 = SAR                                        04735C00 D
0E3F  2F50 CCC0 0C30 C060          EOUTPUT - 1 = CPCR                              04736C0C D
0E40  20D0 00C3 0030 006C          SETCCA - 1 = CPCR      % SET THE CONDITION BITS 04737C00 D
```

```
0E41  56E0 0000 0000 C040              OPCODE - 1 = MPCR
                                                              *
                                                              *
                          OP223:                              * RX TYPE ADD, (R(A))+(Y) -> R(A)
                                                              * SET CC, CARRY, AND OV BITS
0E42  5700 0000 0020 006C              RXMFIELD - 1 = CPCR    * Y INTO B
0E43  4809 0C41 001C 00F0              B L = BR2
0E44  000C 0000 000C 003C              COMP 8 = SAR
0E45  50E0 0000 0000 C060              EMULIN - 1 = CPCR      * (Y) INTO B
0E46  4809 0C41 0030 C0F0              B L = MIR              * (Y) INTO MS WORD OF MIR
0E47  0000 0000 000C C02C              16 = SAR
0E48  4809 E000 0B0C C0F0              A3 = B
0E49  228C 0000 0000 0060              CONTENTSRA - 1 = CPCR  * (R(A)) INTO B
0E4A  4809 0C41 2000 C0F0              B L = A2, BMI
0E4B  000C 0000 000C CC2C              COMP 16 = SAR
0E4C  4809 C040 003C C0F0              A2 + B = MIR           * PERFORM ADDITION
0E4D  7BC9 00C0 0000 C0FC              IF AOV THEN SET LC1
0E4E  1E70 0000 000C 0060              CARRY - 1 = CPCR
0E4F  4F10 0000 0030 C06C              CHECKOV - 1 = CPCR
0E50  4809 C000 000C 00F0              BMI
0E51  4809 0C40 8B3C C0F0              B R = MIR, B
0E52  0000 0000 003C 0020              16 = SAR
0E53  2000 0000 0000 0060              SETCCA - 1 = CPCR      * SET THE CONDITION BITS
0E54  2F5C 0000 0000 0060              EOUTPUT - 1 = CPCR
0E55  56E0 0000 000C 0C40              OPCODE - 1 = MPCR
                                                              *
                                                              *
0E56  5F90 0000 0C00 C060 OPCODE23:  XFCODE - 1 = CPCR        * "F" INTO A2
0E57  4809 C640 0030 C0F0              A2 + AMPCR = AMPCR
0E58  1770 0000 0000 00C0              OP23F - 1 = AMPCR
0E59  4809 0000 003C C0F0              STEP
0E5A  4824 0000 000C C0FC              EXEC
                                                              *
0E5B  1780 C000 0000 0040 OP23F:     OP230 - 1 = MPCR
0E5C  179C 0000 000C C040              OP231 - 1 = MPCR
0E5D  300C 0000 000C C040              FAULT - 1 = MPCR
0E5E  17A0 0000 0000 C040              OP233 - 1 = MPCR
                                                              *
                                                              *
                          OP230:                              * TYPE RR ADD DOUBLE, (R(A)),R(A+1)) +
                                                              * (R(M),R(M+1)) INTO R(A),R(A+1)
                                                              * SET CC, CARRY AND OV BITS
0E5F  4809 2C56 0B3C C0FC              LIT AND B = B          * ISOLATE "M"
0E60  00F0 0000 0C3C 00E0              15 = LIT
0E61  51CC 0000 0000 C060              REGSTACK - 1 = CPCR
0E62  2F3C 0000 0000 0060              EINPUT - 1 = CPCR      * (R(M)) INTO B
0E63  4809 0C41 203C 00FC              B L = A2               * (R(M)) INTO MS WORD OF A2
0E64  001C 0000 0030 C0A0              COMP 16 = SAR ; 1 = LIT
0E65  4809 2F5C 001C C0F0              LIT OR BMAR = MAR2     * ADDR OF R(M+1)
0E66  2F3C 0000 0000 006C              EINPUT - 1 = CPCR      * (R(M+1)) INTO B
0E67  4809 CC53 003C 00F0              A2 OR B = MIR          * MIR = (R(M))/(R(M+1))
0E68  4809 E000 8B3C C0FC              A3 R = B
0E69  30FC 0000 0000 C08C              4 = SAR; 15 = LIT
0E6A  4809 2C56 0BDC 00FC              LIT AND B = B          * ISOLATE "A"
0E6B  51C0 C000 0030 0060              REGSTACK - 1 = CPCR
0E6C  4809 CF41 002C C0FC              BMAR L = BR1           * TEMP STORAGE
0E6D  000C 0000 C00C 003C              COMP 8 = SAR
0E6E  2F3C 0000 0000 C060              EINPUT - 1 = CPCR      * (R(A)) INTO B
0E6F  4809 CC41 2030 00FC              B L = A2               * (R(A)) INTO MS WORD OF A2
0E70  0010 0000 000C C0A0              COMP 16 = SAR ; 1 = LIT
```

```
0E71  4809 2F5C 0C1C 00F0          LIT OR BMAR = MAR2     % R(A+1)
0E72  2F30 C9C0 0000 C060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B          04799C00 D
0E73  4809 CC5C 2000 C0F0          A2 OR B = A2, BMI      % A2 = (R(A))/(R(A+1))     048CCC00 D
0E74  4809 CC40 0C90 C0FC          A2 + B = MIR           % PERFORM ADDITION          048C1000 D
0E75  7BC9 0C03 C030 C0FC          IF AOV THEN SET LC1                               048C2C00 D
0E76  1E70 0000 0030 006C          CARRY - 1 = CPCR                                  04803C00 D
0E77  9F1C 00C3 0000 0060          CHECKOV - 1 = CPCR                                048C4C00 C
0E78  49C9 0000 0D00 C0F0          BMI; SET LC1                                      048C5C00 D
0E79  20D0 C0C3 000C 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS   048C6C00 D
0E7A  4809 0C00 0D3C C0F0          BMI                                               048C7C00 D
0E7B  4809 0C41 2C00 C0F0          B L = A2                                          04808C00 D
0E7C  000C 00C0 0C00 C020          COMP 16 = SAR                                     04809C0C E
0E7D  4809 C000 8C3C 60F0          A2 R = MIR             % (R(A+1))                 0481CC00 D
0E7E  4809 0C40 8B00 00FC          B R = B                % (R(A))                   04811C00 D
0E7F  2F50 C003 CC00 0060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A+1))       04812C00 D
0E80  4809 C0C3 0000 20FC          ASR                    % REF GR1                  04813C00 D
0E81  4809 0F43 801C 00FC          BMAR R = MAR2          % R(A)                     04814C0C D
0E82  0006 0000 0C30 001C          B = SAR                                           04815C00 D
0E83  4809 CC40 0090 00F0          B = MIR                                           04816C00 D
0E84  2F5C 00C3 0030 0060          EOUTPUT - 1 = CPCR     % WRITE NEW R(A)           04817CC0 D
0E85  56E0 00C0 0030 0040          OPCODE - 1 = MPCR                                 04818C00 D
                                                          %                         04819C00 D
                                                          %                         04820C30 D
                                   OP231:                 % RI TYPE 2 ADD DOUBLE     04821C00 D
                                                          % (R(A),R(A+1))+(Y*,Y*+1) INTO   04822C0C D
                                                          % R(A),R(A+1), SET CC, CARRY AND OV BITS 04823C00 D
0E86  2380 0CC0 0C3C 006C          CONTENTSRM - 1 = CPCR                             04824CC0 D
0E87  4809 0C41 CC1C 00F0          B L = BR2              % Y* INTO BR2              04825C00 D
0E88  0000 00C0 0030 0030          COMP 8 = SAR                                      04826C00 D
0E89  60E0 0CC3 000C 0060          EMULIN - 1 = CPCR      % (Y*) INTO B             04827C00 D
0E8A  4809 0C41 2000 00F0          B L = A2               % (Y*) INTO MS WORD OF A2  04828C00 D
0E8B  0C0C 0000 0C30 0020          COMP 16 = SAR                                     04829C0C D
0E8C  48C9 0F47 0C10 00FC          BMAR + 1 L = BR2                                  0483CC0C D
0E8D  000C 0000 CC00 C030          COMP 8 = SAR                                      04831C00 D
0E8E  6CEC 0000 0C0C 006C          EMULIN - 1 = CPCR      % (Y*+1) INTO B           04832C00 D
0E8F  4809 CC5C 0C9C 00F0          A2 OR B = MIR          % MIR = (Y*)/(Y*+1)       04833C00 D
0E90  4809 E0C3 900C 00F0          A3 R = A3                                         04834C00 D
0E91  8CFC 0003 0000 C080          4 = SAR; 15 = LIT                                 04835C00 D
0E92  4809 E1E5 0E70 00F0          A3 AND LIT = B                                    04836C00 D
0E93  51C0 0003 0030 0060          REGSTACK - 1 = CPCR                               04837CC0 D
0E94  48C9 0F41 002C 00FC          BMAR L = BR1           % TEMP STORAGE FOR R(A)    04838CC0 D
0E95  0CCC 00C3 C000 003C          COMP 8 = SAR                                      04839C0C D
0E96  2F3C 02C3 0070 0060          EINPUT - 1 = CPCR      % (R(A)) INTO E           0484CC00 D
0E97  4809 0C41 2000 00F0          B L = A2               % (R(A)) INTO MS WORD OF A2  0484EC00 D
0E98  0010 0003 0000 C0A0          COMP 16 = SAR; 1 = LIT                            04842C00 D
0E99  48C9 2F5C 001C C0F0          LIT OR BMAR = MAR2     % R(A+1)                  0484JC00 D
0E9A  2F30 0CC0 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B          04844C00 D
0E9B  4809 CC5C 2D00 C0F0          A2 OR B = A2, BMI      % A2 = (R(A))/(R(A+1))     04845C00 D
0E9C  48C9 CC40 0090 00FC          A2 + B = MIR           % PERFORM ADDITION          04846C0C D
0E9D  7BC9 0CC0 0C00 00FC          IF AOV THEN SET LC1                               04847C00 D
0E9E  1E70 0003 0030 C060          CARRY - 1 = CPCR                                  04848C00 D
0E9F  9F1C 00C3 0C0C 006C          CHECKOV - 1 = CPCR                                04849CC0 D
0EA0  49C9 0000 0D30 00FC          BMI; SET LC1                                      0485CC00 D
0EA1  20DC 6000 00D0 C060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS   04851C0C D
0EA2  4809 0C00 0D0C C0F0          BMI                                               04852C0C D
0EA3  4809 0C41 2000 C0F0          B L = A2                                          04853C00 D
0EA4  0000 0000 0030 0026          COMP 16 = SAR                                     04854C00 D
0EA5  4809 C000 8030 00FC          A2 R = MIR             % NEW (R(A+1))             04855C0C D
0EA6  4809 CC40 8B00 00F0          B R = B                                           04856C00 D
0EA7  2F5C 00C0 0000 0060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A+1))       04857C00 D
```

```
0EA8  4809 0D0D 0030 20F0          ASR                    % REF BR1                  04856C00 D
0EA9  4809 0F43 8C1C C0FC          BMAR R = MAR2                                     04857000 D
0EAA  00CC 00C0 0030 0010          B = SAR                                           04858000 D
0EA0  4809 0C40 0030 00FC          B = MIR                                           04859000 D
0EAC  2F5C 00C0 009C C060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A))         0485A000 D
0FAD  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                 0485B000 D
                                                          %                          0485C000 D
                                                          %                          0485D000 D
                                   OP233:                 % TYPE RX ADD DOUBLE, (R(A),R(A+1)) +  0485E00C D
                                                          % (Y,Y+1) INTO R(A),R(A+1)  0485F00C D
                                                          % SET CC, CARRY AND OV BITS 04860000 D
0FAE  570C 0000 0000 0060          RXMFIELD - 1 = CPCR    % ADDR OF Y INTO B         04861000 D
0EAF  4809 CC41 0010 00F0          B L = BR2                                         04862000 D
0FB0  000C 00C0 009C 003C          COMP B = SAR                                      04863000 D
0EB1  60E0 00C0 0000 006C          EMULIN - 1 = CPCR      % (Y) INTO B               04864000 D
0ED2  4809 CC41 200C 00F0          B L = A2               % (Y) INTO MS WORD OF A2    04865000 D
0FB3  0C00 00C0 0030 0020          COMP 16 = SAR                                     04866000 D
0EB4  4809 0F47 0C10 00F0          BMAR + 1 L = BR2                                  04867000 D
0EB5  0000 00C0 0000 C030          COMP 0 = SAR                                      04868000 D
0FB6  60E0 00C0 0000 6060          EMULIN - 1 = CPCR      % (Y+1) INTO B             04869000 D
0ED7  48C9 CC5C 0C30 00F0          A2 OR B = MIR          % MIR = (Y)/(Y+1)          0486A000 D
0FB8  4809 E000 8B30 00F0          A3 R = B                                          0486B000 D
0FB9  80FC 0000 C00C 0000          4 = SAR: 15 = LIT                                 0486C000 D
0EBA  4809 2C5C 0B00 00F0          LIT AND B = B                                     0486D000 D
0FDD  51C0 00C0 0C00 C06C          REGSTACK - 1 = CPCR                               0486E000 D
0FDC  48C9 0F41 0C2C C0F0          BMAR L = BR1           % TEMP STORAGE OF R(A)     0486F000 D
0FBD  0C00 00C0 0C30 0030          COMP B = SAR                                      04870000 D
0EBE  2F30 0000 0030 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B            04871000 D
0EBF  4809 CC41 200C 00FC          B L = A2               % (R(A)) INTO MS WORD OF A2 04872000 D
0EC0  0010 00C3 0090 00A0          COMP 16 = SAR#1 = LIT                             04873000 D
0EC1  4809 2F5C 001C C0F0          LIT OR BMAR = MAR2     % R(A+1)                   04874000 D
0EC2  2F3C 0000 000C 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B          04875000 D
0FC3  4809 CC5C 2D9C 00F0          A2 OR B = A2, BMI      % A2 = (R(A))/(R(A+1))     04876000 D
0FC4  48C9 CC40 0030 C0F0          A2 + B = MIR           % PERFORM THE ADDITION      04877000 D
0EC5  7DC9 00C0 0C00 00F0          IF AOV THEN SET LC1                               04878000 D
0FC6  1E7C 00C0 006C C060          CARRY - 1 = CPCR                                  04879000 D
0EC7  4F10 00C0 0C3C 006C          CHECKOV - 1 = CPCR                                0487A00C D
0EC8  49C9 00C0 0DCC C0F0          BMI: SET LC1                                      0487B000 D
0FC9  20DC 00C3 0C3C 006C          SETCCA - 1 = CPCR      % SET THE CONDITION BITS   0487C000 D
0FCA  4809 0000 0DD0 00F0          BMI                                               0487D00C D
0FCB  4809 CC41 203C 0CF0          B L = A2                                          0487E000 D
0ECC  000C 00C0 0000 0020          COMP 16 = SAR                                     0487F000 D
0FCD  4809 CC40 8B30 00F0          B R = B                                           04880000 D
0ECE  4809 C0C3 8C30 60F0          A2 R = MIR             % (R(A+1))                 04881000 D
0ECF  2F5C 00C0 0030 0060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A+1))       04882000 D
0FD0  4809 00C3 009C 20FC          ASR                    % REF BR1                  04883C00 D
0ED1  4809 0F43 8C1C 00F0          BMAR R = MAR2                                     04884000 D
0ED2  0C0C 00C3 0C00 0010          B = SAR                                           04885000 D
0ED3  48C9 0C40 0030 C0F0          B = MIR                % (R(A))                   04886000 D
0FD4  2F50 0000 0000 C060          EOUTPUT - 1 = CPCR                                04887000 D
0ED5  66E0 0000 0030 004C          OPCODE - 1 = MPCR                                 04888000 D
                                                          %                          04889000 D
                                                          %                          0488A000 D
0FD6  5F90 00C0 0000 0060          OPCODE24:  XFCODE - 1 = CPCR  % "F" INTO A2        0488B000 D
0ED7  4809 C643 0C3C 00FC          A2 + AMPCR = AMPCR                                0488C000 D
0ED8  17BC 0000 0000 C0C0          OP24F - 1 = AMPCR                                 0488D000 D
0FD9  4809 00C3 009C 0CF0          STEP                                              0488E000 D
0EDA  4824 00C3 009C 00F0          EXEC                                              0488F000 D
                                                          %                          04890000 D
0EDB  17C0 00C3 0006 0040          OP24F:  OP240 - 1 = MPCR                          04891000 D
```

```
0EDC  1700 0000 0000 0040          OP241 - 1 = MPCR                                  04918C00 D
0EDD  17EC 0000 0000 0040          OP242 - 1 = MPCR                                  04919C00 D
0EDE  17FC 0000 009C 0040          OP243 - 1 = MPCR                                  04920C00 D
                                                    %                                04921C00 D
                                   OP240:           % TYPE RR COMPARE, (R(A))?(R(M))  04922C00 D
                                                    % SET CC, CARRY AND OV BITS       04923C00 D
0EDF  2380 00C0 0C3C 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B             04924C00 D
0EE0  4809 0C40 0030 00F0          B = MIR          % TEMP STORAGE OF (R(M))          04925C00 D
0EE1  4809 E0C0 0B3C 00F0          A3 = B                                            04926C00 D
0EE2  2280 00C0 0C3C 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B             04927C00 D
0EE3  4809 0C41 2D3C 00FC          B L = A2, BMI                                     04928C00 D
0EE4  0C0C 00C0 0030 0020          COMP 16 = SAR                                     04929C00 D
0EE5  4809 0C41 0B3C 06F0          B L = B, MIR     % (R(M)) INTO MS WORD OF B, MIR  04930C00 D
0EE6  200C 0CC0 000C 006C          SETCC - 1 = CPCR  % SET THE CONDITION BITS         04931C00 D
0EE7  4809 00C0 CD0C C0F0          BMI                                               04932C00 D
0EE8  4849 CC5E 0C30 C0F0          A2 - B = MIR; SET LC2                             04933C00 D
0EE9  7BC9 C0C0 000C 00F0          IF AOV THEN SET LC1                               04934C00 D
0EEA  1E7C 00C0 0030 006C          CARRY - 1 = CPCR                                  04935C00 D
0EEB  4F1C 00C0 003C 0060          CHECKOV - 1 = CPCR                                04936C00 D
0EEC  66EC 00C0 00CC 0040          OPCODE - 1 = MPCR                                 04937C00 D
                                                    %                                04938C00 D
                                                    %                                04939C00 D
                                   OP241:           % RI TYPE ? COMPARE, (R(A))?(Y*)  04940C00 D
                                                    % SET CC, CARRY AND OV BITS       04941C00 D
0EED  2280 00C0 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B             04942C00 D
0EEE  4809 0C40 0030 C0F0          B = MIR          % TEMP STORAGE OF (R(A))          04943C00 D
0EEF  238C 00C0 003C 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B             04944C00 D
0EF0  4809 0C41 0D10 00F0          B L = BR2, BMI                                    04945C00 D
0EF1  000C 00C0 0C3C C030          COMP 8 = SAR                                      04946C00 D
0EF2  4809 0C41 2000 00FC          P L = A2         % (R(A)) INTO A2                 04947C00 D
0EF3  00C0 00C3 0000 C020          COMP 16 = SAR                                     04948C00 D
0EF4  6CEC 00C0 000C 0060          EMULIN - 1 = CPCR  % (Y*) INTO B                   04949C00 D
0EF5  4809 0C41 0B30 00F0          P L = B, MIR     % (Y*) INTO MS WORD OF B , MIR   04950C00 D
0EF6  0C0C C0C0 0030 0020          COMP 16 = SAR                                     04951C00 D
0EF7  2C00 00C0 003C 0060          SETCC - 1 = CPCR  % SET THE CONDITION BITS         04952C00 D
0EF8  4809 00C0 0D3C C0FC          BMI              % (Y*) BACK INTO D               04953C00 D
0EF9  4849 CC5E 0030 00FC          A2 - B = MIR; SET LC2                             04954C00 D
0EFA  7BC9 0000 0000 00F0          IF AOV THEN SET LC1                               04955C00 D
0EFB  1E7C 00C0 003C 0060          CARRY - 1 = CPCR                                  04956C00 D
0EFC  4F1C 00C0 00C0 0060          CHECKOV - 1 = CPCR                                04957C00 D
0EFD  66E0 00C0 0030 C04C          OPCODE - 1 = MPCR                                 04958C00 D
                                                    %                                04959C00 D
                                                    %                                04960C00 D
                                   OP242:           % TYPE RK COMPARE, (R(A))?Y       04961C00 D
                                                    % SET CC, CARRY AND OV BITS       04962C00 D
0EFE  4809 2C56 0B3C C0F0          LIT AND B = B                                     04963C00 D
0EFF  00FC 00C0 0C3C 00E0          15 = LIT                                          04964C00 D
0F00  4809 CC52 00C0 00FC          Q EQL B                                           04965C00 D
0F01  6819 00C0 0C3C C0F0          IF TRUE THEN SKIP                                 04966C00 D
0F02  238C 00C0 0000 006C          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B             04967C00 D
0F03  4809 0C41 0B0C C0F0          B L = B                                           04968C00 D
0F04  0000 0000 0C3C C020          COMP 16 = SAR                                     04969C00 D
0F05  4809 EC5C 1C0C C0FC          A3 OR B = A3     % A3 = (R(M))/INSTRUCTION        04970C00 D
0F06  533C 00C0 003C 0060          IFETCH - 1 = CPCR  % *Y* INTO B                    04971C00 D
0F07  4809 E0C0 A03C C0F0          A3 R = A2        % (R(M)) INTO A2                 04972C00 D
0F08  0000 C0C0 0C3C C020          16 = SAR                                          04973C00 D
0F09  4809 0C41 0C30 00F0          A2 + B L = MIR   % Y INTO MS WORD OF MIR          04974C00 D
0F0A  4809 E0C0 CB3C 00F0          A3 = B                                            04975C00 D
0F0B  2280 00C0 0C3C C060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B             04976C00 D
0F0C  4809 0C41 2D3C 00F0          B L = A2, BMI    % (R(A)) INTO MS WORD OF A2      04977C00 D
```

```
0F00  0000 0000 0000 0020          COMP 16 = SAR                                  04978000 D
0F0E  4809 0C40 0030 00F0          0 = MIR                                        04979000 D
0F0F  2000 00C0 0030 C06C          SETCC - 1 = CPCR       % SET THE CONDITION BITS 04980000 D
0F10  4809 0000 0000 00F0          BMI                                            04981000 D
0F11  4849 CC5E 0080 00F0          A2 - B = MIR; SET LC2                          04982000 D
0F12  7BC9 C000 003C C0F0          IF AOV THEN SET LC1                            04983000 D
0F13  1E7C 0003 0030 0060          CARRY - 1 = CPCR                               04984000 D
0F14  4F1C 0003 0030 0060          CHECKOV - 1 = CPCR                             04985000 D
0F15  66E0 0000 0000 0040          OPCODE - 1 = MPCR                              04986000 D
                                                  %                               04987000 D
                                                  %                               04988000 D
                              OP243:              % RX TYPE COMPARE, (R(A)):Y     04989000 D
                                                  % SET CC, CARRY, AND OV BITS    04990000 D
0F16  570C 0000 0000 0060          RXMFIELD - 1 = CPCR    % Y INTO B              04991000 D
0F17  4809 0C41 0010 00F0          B L = BR2                                      04992000 D
0F18  000C 0003 0030 003C          COMP 8 = SAR                                   04993000 D
0F19  60E0 0000 0030 C060          EMULIN - 1 = CPCR      % (Y) INTO B            04994000 D
0F1A  4809 0C41 0030 00F0          B L = MIR                                      04995000 D
0F1B  000C 0000 0030 C020          COMP 16 = SAR                                  04996000 D
0F1C  4809 E0C0 0B00 C0FC          A3 = B                                         04997000 D
0F1D  2280 0000 0000 006C          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B         04998000 D
0F1E  4809 0C41 2000 00F0          B L = A2, BMI                                  04999000 D
0F1F  000C 0003 0030 C020          COMP 16 = SAR                                  05000000 D
0F20  4809 0C40 003C C0F0          A = MIR                                        05001000 D
0F21  200C 00C0 6030 0060          SETCC - 1 = CPCR                               05002000 D
0F22  4809 00C0 0B00 C0F0          BMI                                            05003000 D
0F23  4849 CC5E 0030 00F0          A2 - B = MIR; SET LC2                          05004000 D
0F24  7BC9 0C00 0000 C0FC          IF AOV THEN SET LC1                            05005000 D
0F25  1E7C 00C0 0030 0060          CARRY - 1 = CPCR                               05006000 D
0F26  4F1C 00C3 0030 C060          CHECKOV - 1 = CPCR                             05007000 D
0F27  66E0 00C0 0000 C040          OPCODE - 1 = MPCR                              05008000 D
                                                  %                               05009000 D
                                                  %                               05010000 D
0F28  5F90 00C0 0030 0060     OPCODE25:  XFCODE - 1 = CPCR  % "F" INTO A2          05011000 D
0F29  4809 C643 0040 00F0          A2 + AMPCR = AMPCR                             05012000 D
0F2A  1800 0000 0CC0 00CC          OP25F - 1 = AMPCR                              05013000 D
0F2B  4809 00C0 0030 C0F0          STEP                                           05014000 D
0F2C  4824 00C0 0000 C0F0          EXEC                                           05015000 D
                                                  %                               05016000 D
0F2D  1810 0003 0030 0040     OP25F:   OP250 - 1 = MPCR                           05017000 D
0F2E  182C 0003 0000 0040          OP251 - 1 = MPCR                               05018000 D
0F2F  3000 0003 0030 0040          FAULT - 1 = MPCR                               05019000 D
0F30  183C 0003 0000 C040          OP253 - 1 = MPCR                               05020000 D
                                                  %                               05021000 D
                              OP250:              % TYPE RR COMPARE DOUBLE REGISTER 05022000 D
                                                  % (R(A),R(A+1)) : (R(M),R(M+1)), 05023000 D
                                                  % SET CARRY, CC, AND OV BITS    05024000 D
0F31  4809 0C40 8B00 C0F0          B R = B                                        05025000 D
0F32  80FC 00C3 0C00 C08C          15 = LIT; 4 = SAR                              05026000 D
0F33  4809 2C56 0B0C C0F0          LIT AND B = B          % ISOLATE "A" FIELD     05027000 D
0F34  51CC 0CC0 00CC 006C          REGSTACK - 1 = CPCR                            05028000 D
0F35  2F3C 6003 CC0C C06C          EINPUT - 1 = CPCR      % (R(A)) INTO B         05029000 D
0F36  4809 CC41 2030 00F0          B L = A2                                       05030000 D
0F37  001C 6000 0C30 00A0          COMP 16 = SAR;1 = LIT                          05031000 D
0F38  4809 2F5C 00IC C0FC          LIT OR BMAR = MAR2     % R(A+1)                05032000 D
0F39  2F30 0000 0C0C C060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B       05033000 D
0F3A  4809 CC5C 2030 00F0          A2 OR B = A2           % A2 = (R(A))/(R(A+1)) 05034000 D
0F3B  4809 E0C0 0B30 C0FC          A3 = B                                         05035000 D
0F3C  4809 2C56 0B0C 00F0          LIT AND B = B          % ISOLATE "M" FIELD     05036000 D
0F3D  00FC 0003 0C00 C0E0          15 = LIT                                       05037000 D
```

```
0F3E  51C0 00C0 0000 0060          REGSTACK - 1 = CPCR                                  05038C00 D
0F3F  2F30 00C0 0C30 0060          EINPUT - 1 = CPCR      % (R(M)) INTO B               05039000 D
0F40  9809 0C41 1C30 00F0          B L = A3                                             05040C00 D
0F41  0010 00C0 00F0 00A0          COMP 16 = SAR, 1 = LIT                               05041C00 D
0F42  9809 2F5C 0C1C 00F0          LIT OR BMAR = MAR2     % ADDR OF R(M+1)              05042000 D
0F43  2F30 00C0 0C30 006C          EINPUT - 1 = CPCR      % (R(M+1)) INTO B             05043000 D
0F44  9809 EC5C 0B30 00F0          A3 OR B = B, MIR       % B = (R(M))/(R(M+1))          05044C00 D
0F45  20CC 00C0 0000 0060          SETCC - 1 = CPCR                                     05045C0C D
0F46  9809 00C0 0D3C 00F0          BMI                                                  05046C00 D
0F47  9849 CC5E 0080 00F0          A2 - B = MIR, SET LC2                                05047C00 D
0F48  7BC9 00C0 0000 00F0          IF AOV THEN SET LC1                                  05048C00 D
0F49  1E7C 00C0 0000 0060          CARRY - 1 = CPCR                                     05049C00 D
0F4A  9F1C 0000 0000 0060          CHECKOV - 1 = CPCR                                   0505C0C0 D
0F4B  66E0 00C0 0000 C040          OPCODE - 1 = MPCR                                    05051C00 D
                                                  %                                     05052C00 D
                                                  %                                     05053C00 D
                              OP251:              % RI TYPE 2 COMPARE DOUBLE            05054C0C D
                                                  % (R(A),R(A+1)):(Y*,Y*+1)             05055C00 D
                                                  % SET CC, CARRY, AND OV BITS          05056C00 D
0F4C  9809 0C40 8B3C 00F0          B R = B                                              05057C0C D
0F4D  90FC 0000 003C 0080          4 = SAR, 15 = LIT                                    05058C0C D
0F4E  9809 2C56 0B00 C0F0          LIT AND B = B          % ISOLATE "A" FIELD           05059C00 D
0F4F  51C0 00C0 00CC 0060          REGSTACK - 1 = CPCR                                  05060C00 D
0F50  2F30 00C0 0C3C 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B               05061C00 D
0F51  9809 0C41 2C0C 00F0          B L = A2               % (R(A)) INTO MS WORD OF A2    05062C00 D
0F52  0010 00C0 C0D0 00A0          COMP 16 = SAR , 1 = LIT                              05063C00 D
0F53  9809 2F5C 0C1C 00F0          LIT OR BMAR = MAR2     % R(A+1)                      05064C00 D
0F54  2F30 00C0 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B             05065C00 D
0F55  9809 CC5C 2030 00F0          A2 OR B = A2           % A2 = (R(A))/(RA+1))         05066C0C D
0F56  9809 E156 0B3C 00FC          A3 AND LIT = B         % ISOLATE "M" FIELD           05067C00 D
0F57  00F0 00C0 003C 00EC          15 = LIT                                             05068C00 D
0F58  51C0 00C0 000C C060          REGSTACK - 1 = CPCR                                  05069C0C D
0F59  9809 0F41 0010 00FC          BMAR L = BR2                                         05070C00 D
0F5A  0000 0000 0030 003C          COMP 8 = SAR                                         05071C00 D
0F5B  50E0 0000 0000 0060          EMULIN - 1 = CPCR      % (Y*) INTO B                 05072C00 D
0F5C  9809 0C41 1C30 00F0          B L = A3               % (Y*) INTO MS WORD OF A3      05073C0C D
0F5D  0010 00C0 0C30 C0A0          COMP 16 = SAR , 1 = LIT                              05074C00 D
0F5E  9809 2F5D 001C 00F0          LIT OR BMAR L = MAR2   % ADDR OF Y* + 1              05075C00 D
0F5F  000C 00C0 003C C03C          COMP 8 = SAR                                         05076C0C D
0F60  50E0 C0C0 0030 C060          EMULIN - 1 = CPCR      % (Y*+1) INTO B               05077C00 D
0F61  9809 EC5C 0B30 00F0          A3 OR B = B, MIR       % B = (Y*)/(Y*+1)             05078C00 D
0F62  20CC 00C0 0C0C C06C          SETCC - 1 = CPCR       % SET THE CONDITION BITS      05079C00 D
0F63  9809 C0C0 0D3C 00F0          BMI                                                  05080C0C D
0F64  9849 CC5E 0080 00F0          A2 - B = MIR, SET LC2                                05081C00 D
0F65  7BC9 00C0 0C00 00F0          IF AOV THEN SET LC1                                  05082C00 D
0F66  1E7C C0C0 0000 C06C          CARRY - 1 = CPCR                                     05083C00 D
0F67  9F1C 00C0 0030 C06C          CHECKOV - 1 = CPCR                                   05084C00 D
0F68  66E0 0C00 C0CC C040          OPCODE - 1 = MPCR                                    05085C00 D
                                                  %                                     05086C00 D
                                                  %                                     05087C00 D
                              OP253:              % TYPE RX DOUBLE COMPARE              05088000 D
                                                  %(R(A),R(A+1)):(Y,Y+1)                05089C00 D
                                                  % SET CC, CARRY AND OV BITS           05090C0C D
0F69  5700 00C0 00C0 006C          RXMFIELD - 1 = CPCR     % Y INTO F                  05091C0C D
0F6A  9809 0C41 0010 00FC          D L = BR2                                            05092C00 D
0F6B  000C 00C0 003C C030          COMP 8 = SAR                                         05093C0C D
0F6C  60E0 0000 0030 006C          EMULIN - 1 = CPCR      % (Y) INTO B                  05094C0C D
0F6D  9809 0C41 2C30 C0F0          B L = A2               % (Y) INTO MS WORD OF A2       05095C00 D
0F6E  0010 0000 0030 00A0          COMP 16 = SAR , 1 = LIT                              05096C00 D
0F6F  9809 2F5D 0010 C0F0          LIT OR BMAR L = ER2    % ADDR OF Y+1                 05097C00 D
```

```
0F70  0000 0000 0000 0030          COMP B = SAR                                                  05098000 D
0F71  60E0 0000 0000 0060          ENULIN - 1 = CPCR      % (Y+1) INTO B                         05099000 D
0F72  4809 0050 0080 00F0          A2 OR B = MIR          % MIR = (Y)/(Y+1)                      05100000 D
0F73  4809 0000 9000 00F0          A3 R = A3                                                     05101000 D
0F74  30F0 0000 0000 0080          4 = SAR; 15 = LIT                                             05102000 D
0F75  4809 E156 0030 00F0          A3 AND LIT = B         % ISOLATE "A"                          05103000 D
0F76  5100 0000 0030 0060          REGSTACK - 1 = CPCR                                           05104000 D
0F77  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO E                        05105000 D
0F78  4809 0041 2000 00F0          B L = A2               % (R(A)) INTO MS WORD OF A2             05106000 D
0F79  0010 0003 0000 00A0          COMP 16 = SAR / 1 = LIT                                       05107000 D
0F7A  4809 2F50 0010 00F0          LIT OR BMAR = MAR2     % R(A+1)                               05108000 D
0F7B  2F30 0003 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B                      05109000 D
0F7C  4809 0050 2030 00F0          A2 OR B = A2, DMI      % A2 = (R(A))/(R(A+1))                 05110000 D
0F7D  2000 0000 0000 0060          SETCC - 1 = CPCR       % SET THE CONDITION BITS                05111000 D
0F7E  4809 0000 0030 00F0          BMI                                                           05112000 D
0F7F  4049 0050 0090 00F0          A2 - D = MIR; SET LC2                                         05113000 D
0F80  70C9 0000 0000 00F0          IF ADV THEN SET LC1                                           05114000 D
0F81  1E70 0000 0000 0060          CARRY - 1 = CPCR                                              05115000 D
0F82  4F10 0003 0030 0060          CHECKOV - 1 = CPCR                                            05116000 D
0F83  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                             05117000 D
                                                          %                                      05118000 D
                                                          %                                      05119000 D
0F84  5F90 0000 0070 0060    OPCODE26: XFCODE - 1 = CPCR   % "F" INTO A2                          05120000 D
0F85  4809 0640 0040 00F0          A2 + AMPCR = AMPCR                                            05121000 D
0F86  1840 0000 0000 0000          OP26F - 1 = AMPCR                                             05122000 D
0F87  4809 0000 0030 00F0          STEP                                                          05123000 D
0F88  4824 0000 0000 00F0          EXEC                                                          05124000 D
                                                          %                                      05125000 D
0F89  1850 0003 0000 0040    OP26F:  OP260 - 1 = MPCR                                            05126000 D
0F8A  1860 0003 0000 0040          OP261 - 1 = MPCR                                              05127000 D
0F8B  1870 0000 0030 0040          OP262 - 1 = MPCR                                              05128000 D
0F8C  1880 0003 0000 0040          OP263 - 1 = MPCR                                              05129000 D
                                                          %                                      05130000 D
                                                          %                                      05131000 D
                                   OP260:                 % RR TYPE MULTIPLY (REGISTER)          05132000 D
                                                          % (R(A+1)) X (R(M)) = R(A),R(A+1), SET CC05133000 D
0F8D  4809 0040 8830 00F0          B R = B                                                       05134000 D
0F8E  80F0 0000 0000 0080          4 = SAR; 15 = LIT                                             05135000 D
0F8F  4809 2056 0B00 00F0          LIT AND B = B          % ISOLATE "A"                          05136000 D
0F90  5100 0000 0030 0060          REGSTACK - 1 = CPCR                                           05137000 D
0F91  4809 0F41 0070 00F0          BMAR L = BR1           % TEMP STORAGE                         05138000 D
0F92  0000 0003 0000 0030          COMP B = SAR                                                  05139000 D
0F93  4809 0F45 0B30 00F0          BMAR + 1 = B           % R(A+1)                               05140000 D
0F94  5100 0003 0000 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2                     05141000 D
0F95  2F30 0003 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B                      05142000 D
0F96  4809 0040 2030 00F0          B = A2                                                        05143000 D
0F97  4809 E156 0B00 00F0          A3 AND LIT = B         % ISOLATE "M"                          05144000 D
0F98  00F0 0000 0030 00E0          15 = LIT                                                      05145000 D
0F99  5100 0003 0000 0060          REGSTACK - 1 = CPCR    % R(M) INTO MAR2                       05146000 D
0F9A  2F30 0000 0030 0060          EINPUT - 1 = CPCR      % (R(M)) INTO E                        05147000 D
0F9B  4C6C 0000 0000 0060          MULT - 1 = CPCR        % MULT (R(A+1)) X (R(M))               05148000 D
0F9C  4809 E000 0B30 00F0          A3 = B                 % PRODUCT IN A3                         05149000 D
0F9D  2000 0000 0030 0060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS                05150000 D
0F9E  4809 E000 8C30 00F0          A3 R = MIR             % (R(A))                               05151000 D
0F9F  0000 0003 0000 0020          16 = SAR                                                      05152000 D
0FA0  4809 0000 0000 20F0          ASR                    % REF BR1                              05153000 D
0FA1  4809 0F45 801C 00F0          BMAR R = MAR2          % R(A)                                 05154000 D
0FA2  0000 0003 0000 0010          B = SAR                  \                                    05155000 D
0FA3  2F50 0000 0030 0060          COUTPUT - 1 = CPCR                                            05156000 D
0FA4  4809 0F46 0B20 00F0          BMAR + 1 = D           % R(A+1)                               05157000 D
```

```
0FA5  51C0 0000 0000 0060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2          05156C00 D
0FA6  4809 EC01 1000 00F0          A3 L = A3                                        05159600 D
0FA7  0000 00C0 0C30 0020          COMP 16 = SAR                                    05160C00 D
0FA8  4809 E0C0 8030 C0F0          A3 R = MIR                                       05161800 D
0FA9  2F50 0003 0000 0060          EOUTPUT - 1 = CPCR                               05162400 D
0FAA  66E0 0000 0C0C 0040          OPCODE - 1 = MPCR                                05163000 D
                                                 %                                 05164C30 D
                                                 %                                 05165000 D
                             OP261:              % RI TYPE 2 MULTIPLY (INDIRECT)   05166C00 D
                                                 % (R(A+1)) X (Y*) = R(A),R(A+1), SET CC 05167C00 D
0FAB  4809 CC40 8800 C0F0          B R = B                                          05168C00 D
0FAC  3CFC C0C0 0080 0080          4 = SAR, 15 = LIT                                05169C00 D
0FAD  4809 2C55 C800 00F0          LIT AND B = B         % ISOLATE "A"              05170C00 D
0FAE  51C0 00C0 0C30 0060          REGSTACK - 1 = CPCR                              05171C00 D
0FAF  4809 0F41 0020 C0F0          BMAR L = BR1          % R(A)                     05172C00 D
0FB0  00C0 00C0 C030 003C          COMP 8 = SAR                                     05173C00 D
0FB1  4809 0F46 0E30 C0F0          BMAR + 1 = B          % R(A+1)                   05174C00 D
0FB2  51CC 0000 C030 0060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2         05175C00 D
0FB3  2F30 0FC3 C000 0060          EINPUT - 1 = CPCR     % (R(A+1)) INTO B          05176C00 D
0FB4  4809 0C40 2030 00F0          B = A2                % (R(A+1)) INTO A2          05177C00 D
0FB5  4809 E156 0B00 C0F0          A3 AND LIT = B        % ISOLATE "M"              05178C00 D
0FB6  00F0 00C0 0C3C C3E0          15 = LIT                                         05179C30 D
0FB7  51C0 0000 0C30 C060          REGSTACK - 1 = CPCR   % R(M) INTO MAR2          05180C00 D
0FB8  2F30 C0C0 0C0C C06C          EINPUT - 1 = CPCR     % Y* INTO B                05181C00 D
0FB9  4809 0C41 0C10 C0F0          B L = BR2                                        05182C00 D
0FBA  00C0 00C0 0C00 0030          COMP 8 = SAR                                     05183C00 D
0FBB  60EC 0000 C030 C060          EMULIN - 1 = CPCR     % (Y*) INTO B              05184C00 D
0FBC  4C6C 0000 0C3C 0060          MULT - 1 = CPCR       % MULT (R(A+1)) X (Y*)     05185C00 D
0FBD  48C9 E300 0B00 C0FC          A3 = B                % PRODUCT IN A3             05186C00 D
0FBE  2000 00C0 0C3C 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS   05187C00 D
0FBF  4809 E0C0 8030 C0F0          A3 R = MIR            % (R(A))                   05188C00 D
0FC0  0000 0C03 0000 C020          16 = SAR                                         05189C00 D
0FC1  4809 0000 0000 20FC          ASR                   % REF ER1                  0519CC00 D
0FC2  4809 0F43 801C 0CF0          BMAR R = MAR2         % R(A)                     05191C00 D
0FC3  0000 0CC0 0C30 0010          8 = SAR                                          05192C00 D
0FC4  2F50 0000 0C00 0060          EOUTPUT - 1 = CPCR                               05193C0C D
0FC5  4809 0F46 0B0C 00F0          BMAR + 1 = B          % R(A+1)                   05194000 D
0FC6  51C0 00C0 0C0C 0060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2         05195000 D
0FC7  4809 E001 1000 00FC          A3 L = A3                                        05196C00 D
0FC8  00C0 00C0 0C0C 002C          COMP 16 = SAR                                    05197C00 D
0FC9  4809 E0C0 8030 C0F0          A3 R = MIR                                       05198C00 D
0FCA  2F5C 0003 0C0C 0060          EOUTPUT - 1 = CPCR                               05199C00 D
0FCB  66EC C0C0 0C30 C040          OPCODE - 1 = MPCR                                05200CC0 D
                                                 %                                 05201C00 D
                                                 %                                 05202000 D
                             OP262:              % TYPE RK MULTIPLY (CONSTANT)     05203C00 D
                                                 % (R(A+1)) X Y = R(A),R(A+1), SET CC 05204000 D
0FCC  4809 2C56 CB00 00F0          LIT AND B = B         % ISOLATE "M"              05205C00 D
0FCD  00FC 0003 0000 00CC          15 = LIT*                                        05206CCC D
0FCE  48C9 0C52 0030 00F0          0 EQL B               % CHECK FOP M = 0           05207000 D
0FCF  5819 C000 0C30 C0F0          IF TRUE THEN SKIP                                05208C00 D
0FD0  238C 0000 0C0C C060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B           05209C00 D
0FD1  4809 0C41 0B30 00F0          B L = B                                          05210C0C D
0FD2  00C6 00C0 C030 C02C          COMP 16 = SAR                                    05211C00 D
0FD3  4809 EC5C 1C00 C0F0          A3 OR B = A3          % A3 = (R(M))/INSTRUCTION   05212C00 D
0FD4  6330 0000 0C30 C060          IFETCH - 1 = CPCR     % "Y* INTO B               05213C00 D
0FD5  4809 E0C0 A030 00F0          A3 R = A2             % (R(M)) INTO A2           05214C00 D
0FD6  0000 0000 0003 CC2C          16 = SAR                                         05215C00 D
0FD7  4809 0C40 2C30 C0FC          A2 + B = A2           % Y INTO B                 05216C00 D
0FD8  48C9 E0C0 8B00 00F0          A3 R = B                                         05217C00 D
```

```
0FD9  3CFC 0000 C000 0080          4 = SAR, 15 = LIT
0FDA  4809 2C56 0B90 00FD          LIT AND B = B              % ISOLATE "A"                    05219C00 D
0FDB  51CC 0000 0000 006D          REGSTACK - 1 = CPCR                                         05220000 D
0FDC  4809 0F41 0020 C0FD          BMAR L = BR1               % R(A)                           05221C00 D
0FDD  00CC 0C00 0000 003C          COMP B = SAR                                                05222C00 D
0FDE  4809 0F46 0B00 00FC          BMAR + 1 = B               % R(A+1)                         05223000 D
0FDF  51CC 00C3 0C0C CC60          REGSTACK - 1 = CPCR        % R(A+1) INTO MAR2               05224000 D
0FE0  2F3C 0000 0C00 C06C          EINPUT - 1 = CPCR          % (R(A+1)) INTO B                05225000 D
0FE1  4C60 00C0 0000 0060          MULT - 1 = CPCR            % MULTIPLY (R(A+1)) X  Y         05226C00 D
0FE2  48C9 E0C3 0B0C 00F0          A3 = B                     % PRODUCT IN A3                   05227C00 D
0FE3  2CDC 0CC0 0C00 006C          SETCCA - 1 = CPCR          % SET THE CONDITION BITS          05228C00 D
0FE4  4809 E0CD 8C90 00F0          A3 R = MIR                 % (R(A))                          05229C00 D
0FE5  0000 00C0 C03C 002D          16 = SAR                                                    05230D00 D
0FE6  4809 0000 0C00 20FD          ASR                        % REF BR1                        05231C00 D
0FE7  4809 CF43 8C1C C0F0          BMAR R = MAR2              % R(A)                           05232C00 D
0FE8  000C 0F00 0C00 001C          B = SAR                                                     05233C00 D
0FE9  2F5C 0CC0 0000 C06D          EOUTPUT - 1 = CPCR                                          05234C00 D
0FEA  4809 CF45 0B3C C0F0          BMAR + 1 = B               % R(A+1)                         05235C00 D
0FEB  51CC 0000 0000 006D          REGSTACK - 1 = CPCR        % R(A+1) INTO MAR2               05236000 D
0FEC  4809 E001 1030 C0FC          A3 L = A3                                                   05237C00 D
0FED  0C0C 00C0 0C0C 0C20          COMP 16 = SAR                                               05238C00 D
0FEE  48C9 EC00 8030 00FD          A3 R = MIR                                                  05239C00 D
0FEF  2F50 C0C0 CC3C 0060          EOUTPUT - 1 = CPCR                                          05240C00 D
0FF0  66E0 00C3 0000 0040          OPCODE - 1 = MPCR                                           05241000 D
                                                              %                                05242C00 D
                                                              %                                05243C00 D
                                   OP263:                     % RX TYPE MULTIPLY, (R(A+1)) X (Y) =  05244C00 D
                                                              % R(A),R(A+1), AND SET CC        05245C00 D
0FF1  5700 00C3 0C00 C060          RXMFIELD - 1 = CPCR        % Y INTO B                       05246C00 D
0FF2  4809 CC41 8C1C C0F0          B L = BR2                                                   05247C00 D
0FF3  000C 0C00 0000 0030          COMP B = SAR                                                05248C00 D
0FF4  60E0 00C0 0030 006C          EMULIN - 1 = CPCR          % (Y) INTO B                     05249C00 D
0FF5  48C9 0C40 0030 C0F0          B = MIR                                                     05250C00 D
0FF6  4809 E0C0 8B00 C0F0          A3 R = B                                                    05251C00 D
0FF7  80FC 00C3 0C3C 008D          4 = SAR, 15 = LIT                                           05252000 D
0FF8  4809 2C56 0B90 00F0          LIT AND B = B              % ISOLATE "A"                    05253C00 D
0FF9  51CC 0000 0030 0060          REGSTACK - 1 = CPCR                                         05254000 D
0FFA  4809 0F41 0020 C0FD          BMAR L = BR1               % SAVE R(A)                      05255000 D
0FFB  0000 C000 0C30 0030          COMP B = SAR                                                05256C00 D
0FFC  48C9 0F46 0B00 00FC          BMAR + 1 = B               % R(A+1)                         05257C00 D
0FFD  51C0 0F00 0C00 006C          REGSTACK - 1 = CPCR        % R(A+1) INTO MAR2               05258C00 D
0FFE  2F3C 00C3 0C00 006C          EINPUT - 1 = CPCP          % (R(A+1)) INTO B                05259C00 D
0FFF  4809 0C40 2C00 00F0          B = A2, BMI                                                 05260C00 D
1000  4C6C 00C0 0C30 C060          MULT - 1 = CPCR            % MULTIPLY (R(A+1)) X (Y)        05261C00 D
1001  4809 E0C3 0B00 00F0          A3 = B                     % PRODUCT IN A3                   05262C00 D
1002  20DC 0CC0 0000 0060          SETCCA - 1 = CPCR          % SET THE CONDITION BITS          05263C00 D
1003  4809 E0C3 8C90 C0FD          A3 R = MIR                 % (R(A))                          05264C00 D
1004  00CC C0C3 00C0 0020          16 = SAR                                                    05265C00 D
1005  4809 C0C3 0C0C 20FC          ASR                        % REF BR1                        05266C00 D
1006  4809 0F43 8C1C C0F0          BMAR R = MAR2              % R(A)                           05267C00 D
1007  000C 00C0 0C0C 0C10          B = SAR                                                     05268C00 D
1008  2F50 0CC3 0C30 066C          EOUTPUT - 1 = CPCR                                          05269C00 D
1009  4809 0F46 0B30 C0FC          BMAR + 1 = B               % R(A+1)                         05270C00 D
100A  51C0 00C0 0000 C06C          REGSTACK - 1 = CPCR        % R(A+1) INTO MAR2               05271C00 D
100B  4805 E3C1 1000 00F0          A3 L = A3                                                   05272000 D
100C  4809 E0C0 8C90 00FC          A3 R = MIR                                                  05273C00 D
100D  2F5C 0CC3 0C30 0C60          EOUTPUT - 1 = CPCR                                          05274C00 D
100E  66EC C0C0 0C0C 0040          OPCODE - 1 = MPCR                                           05275C00 D
                                                              %                                05276C00 D
                                                              %                                05277C00 D
```

```
100F  5F90 0000 0000 C060    OPCODE27:  XFCODE - 1 = CPCR    % "F" INTO A2                    05278C00 D
1C10  4809 C640 C040 00F0               A2 + AMPCR = AMPCR                                   05279C00 D
1C11  1890 00C0 0000 C0C0               OP27F - 1 = AMPCR                                    05280C00 D
1C12  4809 00C0 C000 00FC               STEP                                                 05281C00 D
1C13  4824 0000 0000 00F0               EXEC                                                 05282C00 D
                                                                %                            05283C00 D
1C14  18A0 0000 0000 C040    OP27F:     OP270 - 1 = MPCR                                     05284C00 D
1C15  18B0 00C0 0030 C040               OP271 - 1 = MPCR                                     05285C00 D
1C16  18CC 00C0 CC30 C040               OP272 - 1 = MPCR                                     05286000 D
1C17  18DC 00C0 CC30 C04C               OP273 - 1 = MPCR                                     05287CCC D
                                                                %                            05288C00 D
                                                                %                            05289C00 D
                                    OP27C:                      % TYPE RR DIVICE (REGISTER)  05290C00 D
                                                                % R(A),R(A+1)/(R(M)) = R(A+1), REMAINDER 05291C00 D
                                                                % INTO R(A), SET CC AND OV    05292CCC D
1C18  4809 0C40 8D3C 00FC               B R = B                                              05293C0C D
1C19  80FC CCC0 0C00 C08C               4 = SAR; 15 = LIT                                    05294C0C D
1C1A  4809 2C56 0B30 C0FC               LIT AND B = B           % ISOLATE "A"                05295C00 D
1C1B  51CC 00C0 C030 0060               REGSTACK - 1 = CPCR                                  05296C00 D
1C1C  4809 0F41 0C20 00F0               BMAR L = BR1            % TEMP STORAGE               05297C0C E
1C1D  0C0C 00C3 000C CC3C               COMP 8 = SAR                                         05298000 D
1C1E  2F30 0000 0000 0060               EINPUT - 1 = CPCR       % (R(A)) INTO B              05299C00 C
1C1F  4809 0C41 2C70 00F0               B L = A2                                             053C0C00 D
1C20  0C0C 0CC0 0030 0020               COMP 16 = SAR                                        05301C0C D
1C21  4809 CF46 0B3C 00F0               BMAR + 1 = B            % R(A+1)                     0530?C0C D
1C22  51CC 00C0 000C 0060               REGSTACK - 1 = CPCR     % R(A+1) INTO MAR2           05303C00 D
1C23  2F30 0000 0090 C06C               EINPUT - 1 = CPCR       % (R(A+1)) INTO B            0530?C00 D
1C24  4809 CC5C 2070 C0F0               A2 OR B = A2            % A2 = (R(A))/(R(A+1))        05305C00 D
1C25  4809 E156 0B30 C0F0               A3 AND LIT = B          % ISOLATE "M"                 05306F00 D
1C26  00F0 0000 0000 00EC               15 = LIT                                             05307000 D
1C27  51CC 0000 000C C06C               REGSTACK - 1 = CPCR                                  053?8C00 D
1C28  2F30 0CC0 0C30 006C               EINPUT - 1 = CPCR       % (R(M)) INTO B              05309C0C D
1C29  49C9 0C41 0B3C 00F0               B L = B; SET LC1        % LEFT JUSTIFY (R(M)), SET 32 BIT FLAG 05310C00 D
1C2A  D00C 0000 0030 0020               COMP 16 = SAR                                        05311C0C D
1C2B  240C 0000 0000 C060               DIV - 1 = CPCR          % (R(A),R(A+1))/(R(M))       05312000 D
1C2C  2DCB 00C0 003C 0CF0               IF LC1 THEN SET LC1; STEP ELSE SKIP  % CHECK FOR OVERFLOW 05313C0C D
1C2D  50AC 00C0 0030 0060               SETOVBIT - 1 = CPCR     % SET THE OV BIT             05314000 D
1C2E  2819 00C0 C03C 00F0               IF LC1 THEN SKIP                                     05315C00 D
1C2F  5050 0FC0 000C 0060               CLEAROV - 1 = CPCR      % CLEAR THE OV BIT           05316C00 D
1C30  4809 E0C3 0B30 00FC               A3 = B                  % QUOTIENT INTO B            05317C00 E
1C31  4809 C0C0 003C 00F0               A2 = MIR                % REMAINDER INTO MIR          05318C0C D
1C32  20DC 00C0 0C3C C060               SETCCA - 1 = CPCR       % SET THE CONDITION BITS     05319C00 D
1C33  4809 0C00 C03C 20F0               ASR                     % REFERENCE BR1              0532CC0C D
1C34  4809 0F43 8C1C 00FC               BMAR R = MAR2           % R(A)                       05321C00 D
1C35  0000 0000 003C 0010               B = SAR                                              05322C00 D
1C36  2F50 0000 0C0C C060               EOUTPUT - 1 = CPCR      % REMAINDER IN R(A)          05323C0C D
1C37  4809 00C0 0C30 24F0               ASE                     % REFERENCE MAR2             05324C00 D
1C38  4809 0F46 0B0C C0F0               BMAR + 1 = B            % CREATE R(A+1)              05325C0C D
1C39  51CC C0C0 0000 C060               REGSTACK - 1 = CPCR     % R(A+1) INTO MAR2           05326C00 D
1C3A  4809 E0C3 003C 00F0               A3 = MIR                % QUOTIENT                   05327C00 D
1C3B  2F5C 00C0 000C 0060               EOUTPUT - 1 = CPCR                                   05328F0C D
1C3C  66E0 0FC0 0C00 0040               OPCODE - 1 = MPCR                                    05329C00 D
                                                                %                            05330C0C D
                                                                %                            05331C00 D
                                    OP271:                      % PI TYPE 2 DIVIDE (INDIRECT) 05332C00 D
                                                                % (R(A),R(A+1))/(Y*) = R(A+1), REMAINDER 05333C00 D
                                                                % INTO R(A), SET CC AND OV     05334C00 D
1C3D  4809 0C40 8B3C C0F0               B R = B                                              05335C00 D
1C3E  80FC 0000 0000 0080               4 = SAR; 15 = LIT                                    05336C00 D
1C3F  4809 2C56 0B30 C0F0               LIT AND B = B           % ISOLATE "A"                05337C00 D
```

```
1040  51C0 00C0 0030 0060          REGSTACK - 1 = CPCR                                  05338000 D
1041  4809 CF41 0020 C0F0          BMAR L = BR1            % TEMP STORAGE               05339000 D
1042  0C00 C0C0 0C00 C030          COMP 0 = SAR                                         05340000 D
1043  2F30 00C0 0C00 0060          EINPUT - 1 = CPCR       % (R(A)) INTO B              05341000 D
1044  4809 0C41 2000 00FC          B L = A2                                             05342000 D
1045  00CC 0000 0C00 C020          COMP 16 = SAR                                        05343000 D
1046  4809 0F46 0B30 00F0          BMAR + 1 = B            % R(A+1)                     05344000 D
1047  51C0 00C0 0020 0060          REGSTACK - 1 = CPCR     % P(A+1) INTO MAR2           05345000 D
1048  2F30 0CC0 0C00 C060          EINPUT - 1 = CPCR       % (R(A+1)) INTO B            05346000 D
1049  4809 CC50 2C00 00F0          A2 OR 0 = A2            % A2 =(R(A))/(R(A+1))        05347000 D
104A  4809 E156 0B00 00F0          A3 AND LIT = 0          % ISOLATE "M"                05348000 D
104B  00F0 C0C0 000C 00E0          15 = LIT                                             05349000 D
104C  51C0 C0C0 0C30 0060          REGSTACK - 1 = CPCR                                  05350000 D
104D  2F30 00C0 0030 C060          EINPUT - 1 = CPCR       % Y* INTO B                  05351000 D
104E  4809 0C41 001C 0CF0          B L = BR2                                            05352000 D
104F  000C 00C0 0000 C030          COMP 0 = SAR                                         05353000 D
1050  50E0 00C0 0C0C 0060          EMULIN - 1 = CPCR       % (Y*) INTO B                05354000 D
1051  49C9 0C41 0B3C 00F0          B L = B; SET LC1        % LEFT JUSTIFY (Y*)          05355000 D
1052  000C 00C3 0030 0020          COMP 16 = SAR                                        05356000 D
1053  2400 00C0 0C30 C060          DIV - 1 = CPCR          % (R(A),R(A+1))/(Y*)         05357000 D
1054  20CB C0C0 0CC0 C0F0          IF LC1 THEN SET LC1; STEP ELSE SKIP  % CHECK FOR OVEPFLOW  05358000 D
1055  50AC 0000 0C00 0C60          SETOVBIT - 1 = CPCR     % SET THE OV BIT             05359000 D
1056  2819 00C0 0030 00FC          IF LC1 THEN SKIP                                     05360000 D
1057  5050 00C0 C00C 0060          CLEAROV - 1 = CPCR      % CLEAR THE OV BIT           05361000 D
1058  4809 CCC0 0C3C 00FC          A2 = MIR                % REMAINDER INTO MIR          05362000 D
1059  4809 E0C0 0B3C 00FC          A3 = 0                  % QUOTIENT INTO B             05363000 D
105A  20DD 00C0 0C00 0060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS     05364000 D
105B  4809 0CC0 0000 20FC          ASR                     % REF BR1                    05365000 D
105C  48C9 0F46 8C1C 00F0          BMAR R = MAR2           % R(A)                       05366000 D
105D  3000 C0C0 000C C01C          B = SAR                                              05367000 D
105E  2F50 0CC0 000C 0060          EOUTPUT - 1 = CPCR                                   05368000 D
105F  4809 E0C3 0C30 00F0          A3 = MIR                                             05369000 D
1060  48C9 0F46 0B3C 00F0          BMAR + 1 = B            % R(A+1)                     05370000 D
1061  51C0 C000 000C C060          REGSTACK - 1 = CPCR     % R(A+1) INTO MAR2           05371000 D
1062  2F50 C0C0 0C3C 0060          EOUTPUT - 1 = CPCR                                   05372000 D
1063  66E0 00C3 0C3C 0040          OPCODE - 1 = MPCR                                    05373000 D
                                                          %                            05374000 D
                                                          %                            05375000 D
                               OP272:                     % TYPE RK DIVIDE (CONSTANT)   05376000 D
                                                          % (R(A),R(A+1))/Y = R(A+1), REMAINDER  05377000 D
                                                          % INTO R(A), SET CC AND OV    05378000 D
1064  4809 E156 0B00 C0F0          A3 AND LIT = B          % ISOLATE "M"                 05379000 D
1065  00FC 0000 0000 00E0          15 = LIT                                             05380000 D
1066  4809 CC52 0C3C 00F0          0 EQL B                 % CHECK FOR M = 0             05381000 D
1067  6819 C0C0 0C00 C0F0          IF TRUE THEN SKIP                                    05382000 D
1068  238C 00C3 0000 006C          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B               05383000 D
1069  4809 0C41 0B3C 00FC          B L = B                                              05384000 D
106A  0000 0C00 0C3C CC20          COMP 16 = SAR                                        05385000 D
106B  48C9 EC5C 100C 00FC          A3 OR B = A3            % A3 = (R(M))/INSTRUCTION    05386000 D
106C  6330 0CC3 0C00 C060          IFETCH - 1 = CPCR       % "Y" INTO B                 05387000 D
106D  4809 E0C0 A0C0 C0F0          A3 R = A2               % (R(M)) INTO A2              05388000 D
106E  0000 0CC0 0C30 C020          16 = SAR                                             05389000 D
106F  4809 CC41 0B0C 00FC          A2 + B L = B            % Y INTO MS WORD OF B         05390000 D
1070  4809 E0C1 1030 C0F0          A3 L = A3               % CLEAR UHW OF A3             05391000 D
1071  4809 E003 9030 C0F0          A3 R = A3                                            05392000 D
1072  4809 EC5C 100C C0FC          A3 OR B = A3            % A3 = Y/INSTRUCTION          05393000 D
1073  4809 E0C0 8B30 00F0          A3 R = B                                             05394000 D
1074  30FC 00C0 000C 008C          4 = SAR; 15 = LIT                                    05395000 D
1075  4809 2C56 0B00 00F0          LIT AND B = B           % ISOLATE "A"                05396000 D
1076  51C0 0003 C030 0060          REGSTACK - 1 = CPCR                                  05397000 D
```

```
1C77  4809 0F41 0020 00F0          BMAR L = OR1          % TEMP STORAGE                                  05398000 D
1C78  0000 0000 0000 0030          COMP 8 = SAR                                                          05399000 D
1C79  2F30 0000 0000 0060          EINPUT - 1 = CPCR     % (R(A)) INTO B                                 05400000 D
1C7A  4809 0C41 2000 00F0          B L = A2                                                              05401000 D
1C7B  0000 C000 0000 C020          COMP 16 = SAR                                                         05402000 D
1C7C  4809 0F46 0B0C C0F0          BMAR + 1 = B          % R(A+1)                                        05403000 D
1C7D  51CC 0000 0000 0060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2                              05404000 D
1C7E  2F30 0000 0000 0060          EINPUT - 1 = CPCR     % (R(A+1)) INTO B                               05405000 D
1C7F  4809 CC50 2000 00F0          A2 OR B = A2          % A2 = (R(A))/(R(A+1))                          05406000 D
1C80  49C9 E000 8B00 00F0          A3 R = B; SET LC1     % Y INTO B                                      05407000 D
1C81  0000 0000 0000 0020          16 = SAR                                                              05408000 D
1C82  4809 0C41 0B0C C0FC          B L = B               % LEFT JUSTIFY DIVISOR                          05409000 D
1C83  2400 0000 0000 0060          DIV - 1 = CPCR        % (R(A),R(A+1))/Y                               05410000 D
1C84  2BCB 0000 0000 00F0          IF LC1 THEN SET LC1; STEP ELSE SKIP  % CHECK FOR OVERFLOW             05411000 D
1C85  50AC 0000 0000 0060          SETOVBIT - 1 = CPCR   % SET THE OV BIT                                05412000 D
1C86  2819 0000 0000 00F0          IF LC1 THEN SKIP                                                      05413000 D
1C87  5050 0000 0000 0660          CLEAROV - 1 = CPCR    % CLEAR THE OV BIT                              05414000 D
1C88  4809 E000 0B00 C0FC          A3 = B                % QUOTIENT FOR SET CC                           05415000 D
1C89  2600 0000 0000 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS                        05416000 D
1C8A  4809 C000 0030 20F0          A2 = MIR; ASR         % REMAINDER INTO MIR                            05417000 D
1C8B  48C9 0F40 8C1C C0FC          BMAR R = MAR2         % R(A)                                          05418000 D
1C8C  3000 0000 0F00 C010          6 = SAR                                                               05419000 D
1C8D  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR                                                    05420000 D
1C8E  4809 0F46 0B30 00FC          BMAR + 1 = B          % P(A+1)                                        05421000 D
1C8F  51C0 0000 0000 0060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2                              05422000 D
1C90  48C9 E000 0030 00F0          A3 = MIR              % QUOTIENT INTO MIR                             05423000 D
1C91  2F50 0000 0030 0060          EOUTPUT - 1 = CPCR                                                    05424000 D
1C92  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                                     05425000 D
                                                         %                                               05426000 D
                                                         %                                               05427000 D
                                   OP273:                % TYPE RX DIVIDE, (R(A),R(A+1))/(Y) =            05428000 D
                                                         % R(A+1), REMAINDER INTO R(A), SET OV, CC        05429000 D
1C93  570C 0003 0000 0060          RXMFIELD - 1 = CPCR   % Y INTO B                                      05430000 D
1C94  4809 0C41 0010 00F0          B L = BR2                                                             05431000 D
1C95  000C 0003 0030 0030          COMP 8 = SAR                                                          05432000 D
1C96  60E0 0000 0000 C060          EMULIN - 1 = CPCR     % (Y) INTO B                                    05433000 D
1C97  4809 CC40 0C30 C0F0          B = MIR                                                               05434000 D
1C98  4809 E0C0 8B30 00FC          A3 R = B                                                              05435000 D
1C99  30FC 0003 0070 0080          4 = SAR; 15 = LIT                                                     05436000 D
1C9A  48C9 2C56 0B3C 00FC          LIT AND B = B         % ISOLATE "A"                                   05437000 D
1C9B  51CC 0003 0000 C060          REGSTACK - 1 = CPCR   % R(A) INTO MAR2                                05438000 D
1C9C  4809 0F41 002C 00F0          BMAR L = BR1          % TEMP STORAGE                                  05439000 D
1C9D  000C 0000 0030 C030          COMP 8 = SAR                                                          05440000 D
1C9E  2F3C 0000 0000 0060          EINPUT - 1 = CPCR     % (R(A)) INTO B                                 05441000 D
1C9F  4809 0C41 2000 00FC          B L = A2                                                              05442000 D
1CA0  0000 0000 0000 0C20          COMP 16 = SAR                                                         05443000 D
1CA1  4809 0F46 0B2C 00F0          BMAR + 1 = B          % R(A+1)                                        05444000 D
1CA2  51CC 0000 0000 0060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2                              05445000 D
1CA3  2F3C 0000 0000 C060          EINPUT - 1 = CPCR     % (R(A+1)) INTO B                               05446000 D
1CA4  4809 CC5C 2000 C0FC          A2 OR B = A2          % A2 = (R(A))/(R(A+1))                          05447000 D
1CA5  49C9 000C C000 00F0          DMI; SET LC1                                                          05448000 D
1CA6  4809 0C41 0B30 00F0          B L = B               % LEFT JUSTIFY THE DIVISOR                       05449000 D
1CA7  000C 0000 0000 0020          COMP 16 = SAR                                                         05450000 D
1CA8  240C 000C 0000 C060          DIV - 1 = CPCR        % (R(A),R(A+1))/(Y)                             05451000 D
1CA9  2BCB 0000 0030 C0F0          IF LC1 THEN SET LC1; STEP ELSE SKIP  % CHECK FOR OVERFLOW             05452000 D
1CAA  50AC 0C03 0030 C060          SETOVBIT - 1 = CPCR   % SET THE OV BIT                                05453000 D
1CAB  2819 0003 0030 C0F0          IF LC1 THEN SKIP                                                      05454000 D
1CAC  5050 000C C000 C060          CLEAROV - 1 = CPCR    % CLEAR THE OV BIT                              05455000 D
1CAD  4809 E000 0B30 C0FC          A3 = B                % QUOTIENT INTO B                                05456000 D
1CAE  2000 0000 0000 C060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS                        05457000 D
```

```
1CAF  4809 C0C0 0030 20F0          A2 = MIR, ASR          % REMAINDER INTO MIR          05458000 D
1CB0  4809 0F40 B01C C0F0          BMAR R = MAR2          % R(A)                        05459000 D
1CB1  0000 0000 0030 0010          B = SAR                                              05460000 D
1CB2  2F50 00C0 0C30 C060          FOUTPUT - 1 = CPCR     % QUOTIENT INTO R(A+1)        05461000 D
1CB3  4809 0F46 0B30 C0F0          BMAR + 1 = B           % R(A+1)                      05462000 D
1CB4  51C0 00C0 0C0C 0060          REGSTACK - 1 = CPCR    % R(A+1) INTO MAR2            05463000 D
1CB5  4809 E003 0C30 00FC          A3 = MIR               % QUOTIENT INTO MIR            05464000 D
1CB6  2F50 00C0 0C00 C060          FOUTPUT - 1 = CPCR                                   05465000 D
1CB7  66E0 00C0 000C C040          OPCODE - 1 = MPCR                                    05466000 D
                                                          %                            05467000 D
                                                          %                            05468000 D
1CB8  5F90 C0C0 003C 0C60  OPCODE30:  XFCODE - 1 = CPCR   % "F" INTO A2                05469000 D
1CB9  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                   05470000 D
1CBA  18EC 00C3 003C C0CC          OP30F - 1 = AMPCR                                    05471000 D
1CBB  4809 0000 0C00 00FC          STEP                                                 05472000 D
1CBC  4824 0C00 0000 00FC          EXEC                                                 05473000 D
                                                          %                            05474000 D
1CBD  18FC 0C00 C03C 0040  OP30F:  OP300 - 1 = MPCR                                    05475000 D
1CBE  190C 0000 003C C040          OP301 - 1 = MPCR                                     05476000 D
1CBF  191C 0000 0C0C 004C          OP302 - 1 = MPCR                                     05477000 D
1CC0  1920 00C3 0C00 0040          OP303 - 1 = MPCR                                     05478000 D
                                                          %                            05479000 D
                                                          %                            05480000 D
                                   OP300:                 % TYPE RR AND (REGISTER)      05481000 D
                                                          % (R(A)) AND (R(M)) = R(A), SET CC  05482000 D
1CC1  228C 0000 0030 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B               05483000 D
1CC2  4809 0C40 0C90 00F0          B = MIR                % USED AS TEMP                 05484000 D
1CC3  2380 0FF0 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B               05485000 D
1CC4  4809 0C40 2D0C 00F0          B = A2, BMI                                          05486000 D
1CC5  4809 CC56 0B90 C0F0          A2 AND B = MIR, B                                    05487000 D
1CC6  4809 E003 B01C 00FC          A3 R = MAR2            % R(A)                        05488000 D
1CC7  0000 00C0 0030 0020          16 = SAR                                             05489000 D
1CC8  2F50 0000 0000 0C6C          FOUTPUT - 1 = CPCR                                   05490000 D
1CC9  20DC 0000 0000 C06C          SETCCA - 1 = CPCR      % SET THE CONDITION BITS      05491000 D
1CCA  66E0 C0C0 000C 0040          OPCODE - 1 = MPCR                                    05492000 D
                                                          %                            05493000 D
                                                          %                            05494000 D
                                   OP301:                 % RI TYPE 2 AND (INDIRECT)    05495000 D
                                                          % (R(A)) AND (Y*) = R(A), SET CC  05496000 D
1CCB  238C 00C3 0C90 C06C          CONTENTSRM - 1 = CPCR  % Y* INTO B                   05497000 D
1CCC  4809 6C41 001C C0F0          B L = BR2                                            05498000 D
1CCD  0C0C 0CC0 0C20 C030          COMP B = SAR                                         05499000 D
1CCE  6CEC 00C0 C00C 0060          EMULIN - 1 = CPCR      % (Y*) INTO B                 05500000 D
1CCF  4809 0C40 0C8C C0F0          B = MIR                                              05501000 D
1CDC  4809 E0C3 0B00 C0F0          A3 = B                                               05502000 D
1CD1  228C 00C3 0C3C C060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B               05503000 D
1CD2  4809 0C40 2D00 C0F0          B = A2,BMI                                           05504000 D
1CD3  4809 CC56 0B90 00FC          A2 AND B = MIR, B                                    05505000 D
1CD4  2F5C 00C0 C00C 0060          EOUTPUT - 1 = CPCR                                   05506000 D
1CD5  20D0 0CC0 0C3C C06C          SETCCA - 1 = CPCR      % SET THE CONDITION BITS      05507000 D
1CD6  66EC 00C0 0C30 0040          OPCODE - 1 = MPCR                                    05508000 D
                                                          %                            05509000 D
                                                          %                            05510000 D
                                   OP302:                 % TYPE RK AND (CONSTANT)      05511000 D
                                                          % (R(A)) AND Y = R(A), SET CC  05512000 D
1CD7  4809 2C56 0B0C 00F0          LIT AND B = B          % ISOLATE "M"                 05513000 D
1CD8  00FC C0C0 C03C C0EC          15 = LIT                                             05514000 D
1CD9  4809 0C52 0020 0CF0          0 EQL B                % CHECK FOR M = 0              05515000 D
1CDA  6819 0000 0C00 00FC          IF TRUE THEN SKIP                                    05516000 D
1CDB  238C 0003 0C3C C06C          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B               05517000 D
```

```
1CDC  4BC9 0C41 0B0C C0F0          B L = B                                                  05518C00 D
1CDD  0000 0003 0C0C C020          COMP 16 = SAR                                            05519C30 D
1CDE  4809 EC5C 1030 C0F0          A3 OR B = A3          % A3 = CR(M))/INSTRUCTION          0552CC00 D
1CDF  633C 0CC0 C0D0 0060          IFETCH - 1 = CPCR     % "Y" INTO B                       05521C00 D
1CE0  4809 E000 AC0C 0CFC          A3 R = A2                                                05522C00 D
1CE1  0000 C0C0 0030 C020          16 = SAR                                                 05523C00 D
1CE2  4809 CC40 0030 00F0          A2 + B = MIR          % Y INTO MIR                       05524C00 D
1CE3  4809 E0C3 CB30 C0FC          A3 = B                                                   05525C00 D
1CE4  22BC 0CC0 0030 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                   05526C0C D
1CE5  4809 0C40 2D30 C0F0          B = A2, BMI                                              05527C00 D
1CE6  4809 CC56 0B30 00F0          A2 AND B = MIR, B                                        0552BC00 D
1CE7  2F5C 00C0 CC00 0060          EOUTPUT - 1 = CPCR                                       05529C0C D
1CE8  20DC 00C0 C000 C06C          SETCCA - 1 = CPCF     % SET THE CONDITION BITS           05530C00 D
1CE9  66EC CCCC 0030 C040          OPCODE - 1 = MPCR                                        05531C0C C
                                                         %                                  05532C00 D
                                                         %                                  05533C30 D
                              OP3C3:                     % RX TYPE AND                      0553400C D
                                                         % (R(A)) AND (Y) = R(A), SET CC    05535C0C D
1CEA  570C 00C0 0000 006C          PXMFIELD - 1 = CPCR   % Y INTO B                         05536C00 D
1CEB  4809 0C41 0C10 C0F0          B L = BR2                                                05537C00 D
1CEC  0000 C0C0 0C00 C030          COMP B = SAR                                             05538CC0 D
1CED  60E0 0000 CC30 0060          EMULIN - 1 = CPCR     % (Y) INTO B                       05539C0C D
1CEE  4809 0C40 0030 00F0          B = MIR                                                  0554CC00 D
1CEF  4BC9 EC00 0B0C 00F0          A3 = B                                                   05541C00 D
1CF0  22BC 0000 C030 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                   05542000 D
1CF1  4BC9 0C40 2D00 00F0          B = A2, BMI                                              05543C00 D
1CF2  4809 CC56 0B30 00F0          A2 AND B = MIR, B                                        0554400C D
1CF3  2F5C 00C0 0C0C 0060          EOUTPUT - 1 = CPCR                                       05545C30 D
1CF4  20D0 C0C3 CC3C 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS           05546C00 D
1CF5  56E0 00C3 0000 0C40          OPCODE - 1 = MPCR                                        05547C00 D
                                                         %                                  0554B00C D
                                                         %                                  05549C00 D
1CF6  5F9C 0000 0000 C060     OPCODE31:  XFCODE - 1 = CFCR  % "F" INTO A2                   0555CC00 D
1CF7  4809 C640 0030 C0F0          A2 + AMPCR = AMPCR                                       05551C00 D
1CF8  1930 CCF0 CC30 00C0          OP31F - 1 = AMPCR                                        05552C00 D
1CF9  4809 0FC0 0000 00F0          STEP                                                     05553CC0 D
1CFA  4824 00C3 0000 00F0          EXEC                                                     05554C00 D
                                                         %                                  05555C00 D
1CFB  1940 0CC3 0F0C 0040     OP31F:   OP310 - 1 = MPCR                                     05556C00 D
1CFC  1950 00C3 0C30 0040          OP311 - 1 = MPCR                                         05557C00 D
1CFD  1960 CC03 C00C 0040          OP312 - 1 = MPCR                                         05558C00 D
1CFE  1970 00C3 0C30 0040          OP313 - 1 = MPCR                                         05559C00 D
                                                         %                                  0556CCCC D
                                                         %                                  05561C00 D
                              OP310:                     % RR TYPE OR (REGISTER)            05562C00 D
                                                         % (R(A)) OR (R(M)) = R(A), SET CC  05563CC0 D
1CFF  22BC 00C3 0000 006C          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                   05564C00 D
1100  4809 0C40 0030 C0F0          B = MIR                                                  05565C00 D
1101  238C 00C3 0C3C 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                   05566000 D
1102  4809 0C40 2D0C 00F0          B = A2, BMI                                              05567C00 D
1103  4809 CC5C 0B3C 00FC          A2 OR B = MIR, B                                         05568C00 D
1104  4809 E003 8C1C C0FC          A3 R = MAR2           % R(A)                             05569CC0 D
1105  0000 00C0 0030 0020          16 = SAR                                                 0557CC00 D
1106  2F50 00C0 C00C 0060          EOUTPUT - 1 = CPCR                                       05571C00 D
1107  20DC 00C3 C00C 006C          SETCCA - 1 = CPCR     % SET THE CONDITION BITS           0557200C D
1108  56E0 00C3 0C00 C04C          OPCODE - 1 = MPCR                                        05573C0C D
                                                         %                                  05574C00 D
                                                         %                                  05575C00 D
                              OP311:                     % RI TYPE 2 OR (INDIRECT)          05576C0C D
                                                         % (R(A)) OR (Y*) = R(A), SET CC    05577C00 D
```

```
1109  2390 0000 0F00 0060          CONTENTSRM - 1 = CPCR  % Y* INTO B                    05570C00 D
110A  4809 0C41 0010 00F0          B L = BR2                                            05579000 D
110B  000C 0003 0000 0030          COMP 8 = SAR                                         0557BC00 D
110C  50E0 0003 0000 0060          EMULIN - 1 = CPCR      % (Y*) INTO B                 05581C00 D
110D  4809 0C43 0C30 0CF0          B = MIR                                              05582C00 D
110E  4809 EC00 0B3C 00F0          A3 = B                                               0558BC0C E
110F  2280 0003 0000 006C          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B               05584000 D
1110  4809 0C40 2D3C 00FC          B = A2,BMI                                           05585C0C D
1111  4809 CC5C 0B30 00F0          A2 OR B = MIR, B                                     05586C00 D
1112  2F5C 0000 0C30 C060          EOUTPUT - 1 = CPCR                                   05587C00 D
1113  2000 0000 0000 006C          SETCCA - 1 = CPCR      % SET THE CONDITION BITS      05588C00 D
1114  56E0 0000 C030 C040          OPCODE - 1 = MPCR                                    05589C00 D
                                                          %                            0559CC00 D
                                                          %                            05591C00 D
                                   OP312:                 % RK TYPE OR (CONSTANT)       05592C00 D
                                                          % (R(A)) OR Y = R(A), SET (C  05593C00 D
1115  4809 2C55 0B30 00F0          LIT AND B = B          % ISOLATE "M"                 0559AC0C D
1116  00FC 0C00 0C00 C0EC          15 = LIT                                             05595C00 D
1117  4809 0C52 0030 C0F0          0 EQL R                % CHECK FOR M = 0              05596C00 D
1118  6819 00C0 0000 C0F0          IF TRUE THEN SKIP                                    05597C00 D
1119  2380 0000 0020 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B               05598C00 D
111A  4809 0C41 0B30 00F0          B L = B                                              05599C00 D
111B  0000 0000 0030 0020          COMP 16 = SAR                                        0559CC00 D
111C  4809 EC5C 1C30 CCF0          A3 OR B = A3           % A3 = (R(M))/INSTRUCTION     055A1C0C D
111D  6330 0000 0C0C 006C          IFETCH - 1 = CPCR      % "Y" INTO B                  055A2C00 D
111E  4809 E0C0 A0C0 0CFC          A3 R = A2                                            055A3C00 D
111F  00FC 0CC3 0CFC 00A0          16 = SAR; 15 = LIT                                   055A4C0C D
1120  4809 CC40 0C3C 0CF0          A2 + B = MIR           % Y INTO MIR                  055A5C00 D
1121  4809 E0C0 CB00 00F0          A3 = B                                               055A6C00 D
1122  2280 0003 0030 6060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B               055A7CC0 D
1123  4809 0C40 2D3C 0CFC          B = A2, BMI                                          055A8C0C D
1124  4809 CC5C 0B30 00F0          A2 OR B = MIR, B                                     055A9C00 D
1125  2F5C 00C0 0C0C C060          EOUTPUT - 1 = CPCR                                   055B0C00 D
1126  20DC 00C0 C030 C06C          SETCCA - 1 = CPCR      % SET THE CONDITION BITS      055B1C00 D
1127  66E0 0003 0C30 C040          OPCODE - 1 = MPCR                                    055B2C00 D
                                                          %                            055B3C00 D
                                                          %                            055B4C00 D
                                   OP313:                 % RX TYPE OR                  055B5C0C D
                                                          % (R(A)) OR (Y) = R(A), SET CC 055B6C00 D
1128  570C 0000 C03C 0060          RXMFIELD - 1 = CPCR    % Y INTO B                    055B7CCC D
1129  4809 0C41 0010 00F0          B L = BR2                                            055B8C00 D
112A  0C0C 0003 0C30 0C30          COMP 8 = SAR                                         055B9C00 D
112B  60EC 00C3 CC0C 0060          EMULIN - 1 = CPCR      % (Y) INTO B                  055BCCC0 D
112C  4809 0C40 0C3C 00FC          B = MIR                                              055BDC0C D
112D  4809 EC00 0B00 00F0          A3 = B                                               055BEC00 D
112E  2280 0003 0C20 C060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B               055BFC0C D
112F  4809 0C40 2D0C 00F0          B = A2, BMI                                          055C4C00 D
1130  4809 CC5C 0B30 00FC          A2 OR B = MIR, B                                     055C5C0C D
1131  2F50 0C00 0C0C C060          EOUTPUT - 1 = CPCR                                   055C6C00 D
1132  20D0 0000 0C0C C060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS      055C7C00 D
1133  66E0 0003 0030 C040          OPCODE - 1 = MPCR                                    055C8C00 D
                                                          %                            055C9C00 D
                                                          %                            055CCC00 D
1134  5F90 0000 0000 0060          OPCODE32:  XFCODE - 1 = CPCR  % "F" CODE INTO A2     055CDC00 D
1135  4809 C640 0030 00F0          A2 + AMPCR = AMPCR                                   055CEC00 D
1136  198C 0FC0 0000 00C0          OP32F - 1 = AMPCR                                    055CFC00 D
1137  1809 C0C0 0C0C 00FC          STEP                                                 055D0C00 D
1138  4824 0003 0C00 C0F0          EXEC                                                 055D1C00 D
                                                          %                            055D2C0C D
1139  1990 0003 C00C C040    OP32F:  OP32C - 1 = MPCR                                   055D3CC0 D
```

```
113A  19A0 0003 0030 0040          OP321 - 1 = MPCR
113B  19B0 0003 0030 0040          OP322 - 1 = MPCR
113C  19C0 0003 0030 0040          OP323 - 1 = MPCR
                                   %
                                   %
                                   %
                          OP320:                        % RR TYPE EXCLUSIVE OR (REGISTER)
                                                        % (R(A)) XOR (R(M)) = R(A), SET CC
113D  2280 0003 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B
113E  9809 0040 0030 00F0          B = MIR
113F  2380 0003 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B
1140  9809 0040 2000 00FC          B = A2, BMI
1141  9809 0040 0030 00F0          A2 XOR B = MIR, B
1142  9809 0003 8010 00FC          A3 R = MAR2           % R(A)
1143  0000 0000 0020 0020          16 = SAR
1144  2F50 0003 0000 0060          EOUTPUT - 1 = CPCR
1145  2000 0000 0030 006C          SETCCA - 1 = CPCP     % SET THE CONDITION BITS
1146  56E0 0003 0030 0040          OPCODE - 1 = MPCR
                                   %
                                   %
                          OP321:                        % RI TYPE 2 EXCLUSIVE OR (INDIRECT)
                                                        % (R(A)) XOR (Y*) = R(A), SET CC
1147  2380 0003 0030 0060          CONTENTSRM - 1 = CPCR  % Y* INTO B
1148  9809 0041 0010 00F0          B L = BR2
1149  0003 0003 0000 0030          COMP 8 = SAR
114A  50E0 0003 0000 006C          EMULIN - 1 = CPCR     % (Y*) INTO B
114B  9809 0040 0030 00FC          B = MIR
114C  9809 E0C0 0B30 00F0          A3 = B
114D  2280 0003 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B
114E  9809 0040 2000 00F0          B = A2,BMI
114F  9809 0040 0B30 00F0          A2 XOR B = MIR, B
1150  2F50 0003 0000 0060          EOUTPUT - 1 = CPCR
1151  2000 0000 0030 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS
1152  56E0 0003 0030 0040          OPCODE - 1 = MPCR
                                   %
                                   %
                          OP322:                        % RK TYPE EXCLUSIVE OR (CONSTANT)
                                                        % (R(A)) XOR Y = R(A), SET CC
1153  9809 2050 0B30 00FC          LIT AND B = B         % ISOLATE "M"
1154  00FC 0003 0030 00E0          15 = LIT
1155  9809 0052 0000 00F0          0 EQL B               % CHECK FOR M = 0
1156  5819 0003 0000 00F0          IF TRUE THEN SKIP
1157  2380 0003 0000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B
1158  9809 0041 0B10 00F0          B L = B
1159  0000 0003 0030 0020          COMP 16 = SAR
115A  9809 E050 1000 00FC          A3 OR B = A3          % A3 = (R(M))/INSTRUCTION
115B  5330 0003 0000 0060          IFETCH - 1 = CPCR     % "Y" INTO B
115C  9809 E0C0 A030 00F0          A3 R = A2
115D  00F0 0003 0030 00A0          16 = SAR, 15 = LIT
115E  9809 0040 0030 00F0          A2 + B = MIR          % Y INTO MIR
115F  9809 E0C0 0B30 00FC          A3 = B
1160  2280 0003 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B
1161  9809 0040 2000 00FC          B = A2, BMI
1162  9809 0040 0B30 00F0          A2 XOR B = MIR, B
1163  2F50 0003 0000 0060          EOUTPUT - 1 = CPCR
1164  2000 0003 0000 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS
1165  56E0 0000 0000 0040          OPCODE - 1 = MPCR
                                   %
                                   %
                          OP323:                        % RX TYPE EXCLUSIVE OR
                                                        % (R(A)) XOR (Y) = R(A), SET CC
```

```
1166  5700 0000 0000 0060          RXMFIELD - 1 = CPCR   % Y INTO B                    05698000 D
1167  4809 0041 0010 00F0          B L = BR2                                           05699000 D
1168  0000 0000 0000 0030          COMP B = SAR                                        05700000 D
1169  60EC 0000 0000 0060          EMULIN - 1 = CPCR     % (Y) INTO B                  05701000 D
116A  4809 0C40 0090 00F0          B = MIR                                             05702000 D
116B  4809 E000 CB00 C0F0          A3 = B                                              05703000 D
116C  22B0 0000 0000 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B              05704000 D
116D  4809 0040 2D00 00F0          B = A2, BMI                                         05705000 D
116E  4809 0090 0B3C 00F0          A2 XOR B = MIR, B                                   05706000 D
116F  2F50 0000 0000 C060          FOUTPUT - 1 = CPCR                                  05707000 D
1170  200C 0000 000C C060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS      05708000 D
1171  66EC 0000 0090 0040          OPCODE - 1 = MPCR                                   05709000 D
                                                         %                             05710000 D
                                                         %                             05711000 D
1172  5F90 0000 0000 C060    OPCODE33:  XFCODE - 1 = CPCR  % "F" INTO A2               05712000 D
1173  4809 0640 0040 00F0          A2 + AMPCR = AMPCR                                  05713000 D
1174  19DC 000D 0000 00CC          OP33F - 1 = AMPCR                                   05714000 D
1175  4809 0000 000F C0FC          STEP                                                05715000 D
1176  4824 C000 0C0C 00FC          EXEC                                                05716000 D
                                                         %                             05717000 D
1177  19EC 000D 0000 0040    OP33F:  OP330 - 1 = MPCR                                  05718000 D
1178  19F0 0000 0000 0040          OP331 - 1 = MPCR                                    05719000 D
1179  1A0C 000D 0C00 0040          OP332 - 1 = MPCR                                    05720000 D
117A  1A10 000D 0C00 0040          OP333 - 1 = MPCR                                    05721000 D
                                   OP330:                % TYPE RR MASKED SUBSTITUTE (REGISTER)  05722000 D
                                                         % IF (R(A+1))N = 1, (R(M)) INTO R(A)N, CC05723000 D
117B  2380 000D 000C D060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B              05724000 D
117C  4809 0C40 0C90 C0F0          B = MIR               % TEMP STORAGE               05725000 D
117D  4809 E0CD 8B30 00FC          A3 R = B                                            05726000 D
117E  30FC C0C0 0000 0080          4 = SAR; 15 = LIT                                   05727000 D
117F  4809 2056 0B00 C0F0          LIT AND B = B         % ISOLATE "A"               05728000 D
1180  51C0 000D 0C0C 0060          REGSTACK - 1 = CPCR   % R(A) INTO MAR2             05729000 D
1181  48C9 0F41 0C20 00F0          BMAR L = BR1          % STORE R(A)                05730000 D
1182  00C0 000D 0C00 D03C          COMP B = SAR                                        05731000 D
1183  4809 0F46 0B0C 00F0          BMAR + 1 = B          % R(A+1)                    05732000 D
1184  51C0 000D 0C00 0060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2           05733000 D
1185  2F3C 000D 0000 C060          EINPUT - 1 = CPCR     % (R(A+1)) INTO B           05734000 D
1186  4809 0040 2D00 00F0          B = A2, BMI                                         05735000 D
1187  4809 C056 009C 00F0          A2 AND B = MIR        % (R(A+1)) AND (R(M))       05736000 D
1188  4809 0000 0009 20FC          ASR                   % REFERENCE BR1             05737000 D
1189  4809 0F40 B01C 00F0          BMAR R = MAR2                                       05738000 D
118A  00C0 C00D 0000 C010          B = SAR                                             05739000 D
118B  2F3C 0000 0000 C060          EINPUT - 1 = CPCR     % (R(A)) INTO B             05740000 D
118C  48C9 0040 2D0C CCF0          B = A2,BMI                                          05741000 D
118D  4809 CC50 0B30 00F0          A2 OR B = MIR, B                                    05742000 D
118E  2F5C 0000 C000 0060          FOUTPUT - 1 = CPCR                                  05743000 D
118F  200C CCC0 0000 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS      05744000 D
1190  66EC 0000 0000 C040          OPCODE - 1 = MPCR                                   05745000 D
                                                         %                             05746000 D
                                                         %                             05747000 D
                                   OP331:                % RI TYPE 2 MASKED SUBSTITUTE (INDIRECT) 05748000 D
                                                         % IF (R(A+1))N=1, (Y*)N = R(A)N, SET CC 05749000 D
1191  238C 0000 0000 C060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B             05750000 D
1192  4809 0041 D010 00F0          B L = BR2                                           05751000 D
1193  0000 0000 0000 0030          COMP B = SAR                                        05752000 D
1194  60EC 0000 000C 0060          EMULIN - 1 = CPCR     % (Y*) INTO B               05753000 D
1195  4809 CC40 0C30 C0F0          B = MIR               % TEMP STORAGE               05754000 D
1196  4809 E0C0 8B3C C0FC          A3 R = B                                            05755000 D
1197  30FC 000D 0C00 0080          4 = SAR; 15 = LIT                                   05756000 D
1198  4809 2056 0B0C C0FC          LIT AND B = B         % ISOLATE "A"               05757000 D
```

```
1199  51C0 C0C0 0C0C 0060          REGSTACK - 1 = CPCR     % R(A) INTO MAR2                05758C00 D
119A  4809 0F41 0070 00FC          BMAR L = BR1            % TEMP STORAGE                  05759C00 D
119B  000C 00C0 0030 0030          COMP B = SAR                                            05760C00 D
119C  4809 0F45 0B30 C0F0          BMAR + 1 = B            % R(A+1)                        05761C00 D
119D  51C0 00C0 003C 0060          REGSTACK - 1 = CPCR     % R(A+1) INTO MAR2              05762C00 D
119E  2F30 0000 C00C 006C          EINPUT - 1 = CPCR       % (R(A+1)) INTO B               05763000 D
119F  4809 0C40 2D00 00F0          D = A2, BMI                                             05764000 D
11A0  4809 CC56 0030 C0F0          A2 AND B = MIR          % (R(A+1)) AND (Y*)             05765C00 D
11A1  4809 0C00 0C00 20F0          ASR                     % REFERENCE DR1                 05766000 D
11A2  4809 0F43 801C 00F0          BMAR R = MAR2           % R(A)                          05767C00 D
11A3  000C C000 000C 001C          B = SAR                                                 05768C00 D
11A4  2F30 0000 0000 C060          EINPUT - 1 = CPCR       % (R(A)) INTO B                 05769C00 D
11A5  4809 0C40 2D3C C0F0          B = A2,BMI                                              0577CC0C D
11A6  4809 CC5C 0B30 00FC          A2 OR B = MIR, B                                        05771C00 E
11A7  2F50 C0C0 0C3C C06C          EOUTPUT - 1 = CPCR                                      05772C00 D
11A8  20DC 00C0 CC0C 0060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS        05773C0C D
11A9  66EC 00C0 0C00 0040          OPCODE - 1 = MPCR                                       05774CD0 E
                                                           %                               05775C0C D
                                                           %                               05776C00 E
                                   0P332:                  % RK TYPE MASKED SUBSTITUTE (CONSTANT)  05777000 D
                                                           % IF (R(A+1))N= 1, YN INTO R(A)N, SET CC 05778C00 D
11AA  4809 2C56 0B3C 00F0          LIT AND B = B           % ISOLATE "M"                   05779C00 D
11AB  00FC C000 C030 C0L0          15 = LIT                                                0578CC00 E
11AC  4809 0C52 0030 C0F0          0 EQL B                 % CHECK FOR M = 0               05781C00 D
11AD  6819 00C0 0C0C 00FC          IF TRUE THEN SKIP                                       05782000 D
11AE  2380 0CC0 0C0C 0060          CONTENTSRM - 1 = CPCR   % (R(M)) INTO B                 0578300C D
11AF  4809 0C41 0B30 C0FC          B L = B                                                 0578400C D
11B0  000C 00C0 003C 002C          COMP 16 = SAR                                           0578500C D
11B1  4809 EC5C 1000 C0F0          A3 OR B = A3            % A3 = (R(M))/INSTRUCTION       05786C30 D
11B2  633C 00C0 003C C060          IFETCH - 1 = CPCR       % "Y" INTO B                    05787CC0 D
11B3  4809 ECC3 A000 00F0          A3 R = A2               % (R(M)) INTO A2                05788C00 D
11B4  000C C0C0 0C3C 0020          16 = SAR                                                05789C00 D
11B5  4809 CC40 0C30 C0F0          A2 + B = MIR            % Y INTO MIR                    0579CC00 E
11B6  4809 E0C3 8B0C C0F0          A3 R = B                                                05791C00 D
11B7  80FC 00C3 0C70 00EC          4 = SAR, 15 = LIT                                       05792C00 D
11B8  4809 2C56 0B00 0CF0          LIT AND B = B           % ISOLATE "A"                   05793C00 D
11B9  51C0 0000 00C0 C060          REGSTACK - 1 = CPCR     % R(A) INTO MAR2                05794CC0 D
11BA  48C9 0F41 0C2C 00FC          BMAR L = BR1            % TEMP STORAGE                  05795GC0 D
11BB  0CCC 0C00 0C30 0030          COMP B = SAR                                            05796C00 D
11BC  4809 0F46 0B3C 00F0          BMAR + 1 = B            % R(A+1)                        05797000 D
11BD  51C0 C000 000C 006C          REGSTACK - 1 = CPCR     % R(A+1) INTO MAR2              05798C00 D
11BE  2F30 00C0 0C00 0060          EINPUT - 1 = CPCR       % (R(A+1)) INTO B               05799C00 D
11BF  4809 CC43 2D00 00F0          B = A2, BMI                                             058CC000 D
11C0  4809 CC56 0C3C 00F0          A2 AND B = MIR          % (R(A+1)) AND Y                05801C00 D
11C1  4809 C000 0C00 20FC          ASR                     % REFERENCE BR1                 05802C00 D
11C2  4809 0F43 801C C0F0          BMAR R = MAR2           % R(A)                          058C3C0C D
11C3  0000 0CC3 0C3C C010          B = SAR                                                 058C4C3C D
11C4  2F30 CC00 0030 C060          EINPUT - 1 = CPCR       % (R(A)) INTO B                 058C5C00 D
11C5  4809 0C40 2D30 C0FC          B = A2,BMI                                              058C6C00 D
11C6  48C9 CC5C 0B30 00F0          A2 OR B = MIR, B                                        058C7C00 D
11C7  2F5C 00C0 0C00 C060          EOUTPUT - 1 = CPCR                                      058C8C00 D
11C8  20DC 00C0 00CC 0060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS        05809C00 D
11C9  66E0 0CC0 00C0 0040          OPCODE - 1 = MPCR                                       0581CC0C D
                                                           %                               05811C0C D
                                                           %                               05812C00 D
                                   0P333:                  % RX TYPE MASKED SUBSTITUTE     05813C00 E
                                                           % IF (R(A+1))N= 1, (Y)N INTO R(A), SET CC05814C0C D
11CA  570C C0C3 0C0C 0060          RXMFIELD - 1 = CPCR     % Y INTO D                      05815C00 D
11CB  4809 CC41 C010 C0F0          D L = BR2                                               05816CCC D
11CC  0000 0C00 0000 0030          COMP B = SAR                                            05817C00 E
```

```
11CD  60E0 0000 0000 0060          EMULIN - 1 = CPCR     % (Y) INTO B                  05818000 D
11CE  4809 0C40 0090 C0F0          0 = MIR               % TEMP STORAGE                 05819000 D
11CF  4809 E0C0 8B30 00F0          A3 R = B                                            05820000 D
11D0  80F0 CC00 0000 0080          4 = SAR; 15 = LIT                                   05821000 D
11D1  4809 2C56 0B00 C0F0          LIT AND B = B         % ISOLATE "A"                 05822000 D
11D2  51C0 0000 0000 C060          REGSTACK - 1 = CPCR   % R(A) INTO MAR2               05823000 D
11D3  4809 0F4E 0020 00F0          BMAR L = BR1          % TEMP STORAGE                05824000 D
11D4  0000 C0C0 0C0C 0030          COMP B = SAR                                        05825000 D
11D5  4809 0F46 0B30 00FC          BMAR + 1 = B          % R(A+1)                      05826000 D
11D6  51C0 00C0 0000 006C          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2            05827000 D
11D7  2F30 0000 0090 0060          EINPUT - 1 = CPCR     % (R(A+1)) INTO B             05828000 D
11D8  4809 0C40 2D0C 00F0          0 = A2, BMI                                         05829000 D
11D9  4809 CC56 003C C0F0          A2 AND B = MIR        % (R(A+1)) AND (Y)            05830000 D
11DA  4809 0000 0000 20F0          ASR                   % REFERENCE BR1               05831000 D
11DB  4809 0F4D 8C1C 00FC          BMAR R = MAR2                                       05832000 D
11DC  0000 00C0 0000 0010          B = SAR                                             05833000 D
11DD  2F30 00C0 0090 C060          EINPUT - 1 = CPCR     % (R(A)) INTO D               05834000 D
11DE  4809 0C40 2D0C 00F0          B = A2,BMI                                          05835000 D
11DF  4809 CC5C 0B30 00F0          A2 OR B = MIR, B                                    05836000 D
11E0  2F50 00C0 0000 0060          EOUTPUT --1 = CPCR                                  05837000 D
11E1  2000 0000 0000 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS      05838000 D
11E2  66E0 0000 009C 0040          OPCODE - 1 = MPCR                                   05839000 D
                                                         %                             05840000 D
                                                         %                             05841000 D
11E3  5F90 0000 0000 0060    OPCODE34:  XFCODE - 1 = CPCR % "F" INTO A2                05842000 D
11E4  4809 C640 0040 00F0          A2 + AMPCR = AMPCR                                  05843000 D
11E5  1A20 00C0 0090 00C0          OP34F - 1 = AMPCR                                   05844000 D
11E6  4809 0000 0000 00F0          STEP                                                05845000 D
11E7  4824 0000 0090 00F0          EXEC                                                05846000 D
                                                         %                             05847000 D
11E8  1A30 0000 0090 0040    OP34F:  OP340 - 1 = MPCR                                  05848000 D
11E9  1A40 0000 0090 0040          OP341 - 1 = MPCR                                    05849000 D
11EA  1A50 0000 0090 0040          OP342 - 1 = MPCR                                    05850000 D
11EB  1A60 00C0 0090 C040          OP343 - 1 = MPCR                                    05851000 D
                                                         %                             05852000 D
                                                         %                             05853000 D
                                   OP340:                % TYPE RR COMPARE MASKED (REGISTER)  05854000 D
                                                         % (R(A)) AND (R(A+1)) : (R(M)) AND  05855000 D
                                                         % (R(A+1)), SET CC            05856000 D
11EC  2380 0000 0090 C060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B              05857000 D
11ED  4809 0C40 0090 00F0          0 = MIR               % TEMP STORAGE                 05858000 D
11EE  4809 E0C0 8B20 C0F0          A3 R = B                                            05859000 D
11EF  80F0 0000 0090 C080          4 = SAR; 15 = LIT                                   05860000 D
11F0  4809 2C56 0B30 00FC          LIT AND B = B         % ISOLATE "A"                 05861000 D
11F1  51C0 0000 0090 C06C          REGSTACK - 1 = CPCR                                 05862000 D
11F2  2F30 0000 0000 C060          EINPUT - 1 = CPCR     % (R(A)) INTO B               05863000 D
11F3  4809 0C40 2030 00F0          B = A2                                              05864000 D
11F4  4809 0F46 0B20 C0F0          BMAR + 1 = B          % R(A+1)                      05865000 D
11F5  51C0 0000 0090 6060          REGSTACK - 1 = CPCR   % R(A+1) INTO MAR2           05866000 D
11F6  2F30 0000 0090 0060          EINPUT - 1 = CPCR     % (R(A+1)) INTO B             05867000 D
11F7  4809 CC56 2000 00F0          A2 AND B = A2         % (R(A)) AND (R(A+1))         05868000 D
11F8  4809 0C40 1D30 C0F0          B = A3, BMI                                         05869000 D
11F9  4809 EC56 0B20 00F0          A3 AND B = B          % (R(M)) AND (R(A+1))         05870000 D
11FA  2000 0000 0090 C060          SETCC - 1 = CPCR      % SET THE CONDITION BITS      05871000 D
11FB  66E0 0000 0090 C040          OPCODE - 1 = MPCR                                   05872000 D
                                                         %                             05873000 D
                                                         %                             05874000 D
                                   OP341:                % R1 TYPE 2 COMPARE MASKED(INDIRECT)  05875000 D
                                                         % (R(A)) AND (R(A+1)) : (Y*) AND (R(A+1))05876000 D
                                                         % AND SET CC                  05877000 D
```

```
11FC  2380 0000 0030 0060          CONTENTSRM - 1 = CPCR   % Y= INTO B                  05878C00 D
11FD  4809 0C41 0010 00FD          B L = BR2                                            05879C0C D
11FE  0C00 CCC0 0030 C030          COMP 8 = SAR                                         05880C00 D
11FF  60E0 00C0 0030 0060          EMULIN - 1 = CPCR       % (Y=) INTO B                05881C00 D
1200  4809 0C40 0030 00FC          B = MIR                 % TEMP STORAGE                05882C00 D
1201  4809 E000 8800 00F0          A3 R = B                                             05883C00 D
1202  80FC 0003 C00C C080          4 = SAR, 15 = LIT                                    05884C00 D
1203  4809 2C56 0B30 0CFC          LIT AND B = B           % ISOLATE "A"                05885C0C D
1204  51CC 0000 C000 006D          REGSTACK - 1 = CPCR                                  05886C00 D
1205  2F30 0C00 0030 C06D          EINPUT - 1 = CPCR       % (R(A)) INTO E              05887C00 D
1206  4809 0C40 2C30 C0F0          B = A2                                               05888CCC D
1207  48C9 0F45 0D30 00F0          BMAR + 1 = B            % R(A+1)                     05889C00 D
1208  51CC C000 C000 006D          REGSTACK - 1 = CPCR     % R(A+1) INTO MAR2           0589CC00 D
1209  2F3C 0C00 0C00 0060          EINPUT - 1 = CPCR       % (R(A+1)) INTO B            05891C00 D
120A  4809 CC56 2030 C0FC          A2 AND B = A2           % (R(A)) AND (R(A+1))        05892C0C D
120B  4809 0C40 1D0C C0FC          B = A3, BMI                                          05893C00 D
120C  4809 EC56 0D30 CCFD          A3 AND B = D            % (Y=) AND (R(A+1))          05894C00 D
120D  200C 0CC0 CC0C 0060          SETCC - 1 = CPCR        % SET THE CONDITION BITS     05895C00 D
120E  66E0 CCC0 0C00 C040          OPCODE - 1 = MPCR                                    05896C00 D
                                                           %                            05897C00 D
                                                           %                            05898C00 D
                             OP342:                        % TYPE RK COMPARE MASKED (CONSTANT)   05899C00 D
                                                           % (R(A)) AND (R(A+1)) : Y AND (R(A+1)),  0590CC00 D
                                                           % AND SET CONDITION BITS      05901C00 D
120F  4809 2C56 0E00 C0FD          LIT AND B = B           % ISOLATE M                   0590C0C0 D
1210  00F0 0000 0000 00EC          15 = LIT                                             0590C3C0C D
1211  4809 0C52 0000 00F0          0 EQL B                 % CHECK FOR M = 0             0590C4C0 D
1212  6819 0000 0030 00FD          IF TRUE THEN SKIP                                    0590C5C00 D
1213  2380 0000 0030 C06C          CONTENTSRM - 1 = CPCR   % (R(M)) INTO B              0590C6C00 D
1214  4809 0C41 0B20 00FD          B L = B                                              05907C00 D
1215  0000 C0C0 0030 0020          COMP 16 = SAR                                        0590CCDD D
1216  4809 EC5C 1000 0DFD          A3 OR B = A3            % A3 = (R(M))/INSTRUCTION     05909000 D
1217  5330 0003 0006 0060          IFETCH - 1 = CPCR       % "Y" INTO B                 0591D0D0 D
1218  4809 EC0C A000 C0F0          A3 R = A2               % (R(M)) INTO A2              05911C0C D
1219  0000 00C3 000C D020          16 = SAR                                             05912C0C D
121A  48C9 CC40 C030 00F0          A2 + B = MIR            % Y INTO D                    05913C00 D
121B  4809 E0CC 88BC C0F0          A3 R = B                                             05914C0C D
121C  90FC 0000 0020 C080          4 = SAR, 15 = LIT                                    05915C00 D
121D  4809 2C56 0B30 0CF0          LIT AND B = B           % ISOLATE "A"                05916C00 D
121E  51CC C0C3 000C 006C          REGSTACK - 1 = CPCR     % R(A) INTO MAR2             05917C00 D
121F  2F30 0CC0 0C0C 006C          EINPUT - 1 = CPCR       % (R(A)) INTO E              05918C00 D
1220  4809 0C40 2C3C C0F0          B = A2                                               05919C00 D
1221  4809 CF45 0B30 C0FC          BMAR + 1 = B            % R(A+1)                     0592CC00 D
1222  51CC 0CC3 003C 006C          REGSTACK - 1 = CPCR     % R(A+1) INTO MAR2           05921C30 D
1223  2F3C 0CC3 0020 006C          EINPUT - 1 = CPCR       % (R(A+1)) INTO B            05922C0C D
1224  4809 CC56 203C 00F0          A2 AND B = A2           % (R(A)) AND (F(A+1))        05923C00 D
1225  4809 0C4D 1D0C 00F0          B = A3, BMI                                          05924C00 D
1226  4809 EC56 0B30 C0F0          A3 AND B = B            % Y AND (R(A+1))             05925C00 D
1227  200C 0C00 000C 0D60          SETCC - 1 = CPCR        % SET THE CONDITION BITS     05926C00 D
1228  66E0 0000 0030 C040          OPCODE - 1 = MPCR                                    05927C00 D
                                                           %                            05928C00 D
                                                           %                            05929C0C D
                             OP343:                        % RX TYPE COMPARE MASKED      05930C00 D
                                                           % (R(A)) AND (R(A+1)) : (Y) AND (R(A+1))  05931C00 D
                                                           % AND SET CONDITION BITS      05932C00 D
1229  570C 0000 0030 C06C          RXMFIELD - 1 = CPCR     % Y INTO B                   05933CCC D
122A  4809 0C41 0010 00FC          B L = BR2                                            05934C0C D
122B  0000 0CC0 0C00 C030          COMP 8 = SAR                                         05935C0C D
122C  60E0 0000 0000 006C          EMULIN - 1 = CPCR       % (Y) INTO B                 05936C00 D
122D  4809 0C40 0030 00FC          B = MIR                 % TEMP STORAGE                05937CC0 D
```

```
122E  4809 E000 8B30 00FC              A3 R = B                                                  05938C0C D
122F  80F0 0003 0000 0080              4 = SAR; 15 = LIT                                         05939C00 D
1230  48C9 2C56 0B0C 00FC              LIT AND B = B              % ISOLATE "A"                  05940CC0 D
1231  51C0 00C3 0030 0060              REGSTACK - 1 = CPCR        % R(A) INTO MAR2               05941C0C D
1232  2F3C 00C3 0000 0060              EINPUT - 1 = CPCR          % (R(A)) INTO E                05942C00 D
1233  4809 0C40 2C0C C0F0              B = A2                                                    05943C00 D
1234  4809 0F46 0B3C CCFC              BMAR + 1 = B               % R(A+1)                       05944C00 D
1235  51C0 0000 0000 0060              REGSTACK - 1 = CPCR        % R(A+1) INTO MAR2             05945C0C D
1236  2F3C 0000 0C20 0060              EINPUT - 1 = CPCR          % (R(A+1)) INTO B              05946C00 D
1237  4809 CC56 203C 00FC              A2 AND B = A2              % (R(A)) AND (R(A+1))          05947C00 D
1238  4809 CC40 1030 00F0              B = A3, BMI                                               05948C0C D
1239  4809 EC56 0F30 00F0              A3 AND B = B               % (Y) AND (R(A+1))             05949C00 D
123A  2000 0000 0C3C C060              SETCC - 1 = CPCR           % SET THE CONDITION BITS       05950C00 D
123B  66E0 00C3 0000 C040              OPCODE - 1 = MPCR                                         05951C3C D
                                                                  %                             05952C00 D
123C  5F90 00C0 0C20 0060      OPCODE35:  XFCODE - 1 = CPCR      % "F" INTO A2                  05953C0C D
123D  4809 C640 0C40 00FC              A2 + AMPCR = AMPCR                                        05954C0C D
123E  1A70 00C0 0C30 C0CC              OP35F- 1 = AMPCR                                          05955C3C D
123F  4809 00C3 0000 00FC              STEP                                                      05956C00 D
1240  4824 0CC0 0C00 C0FC              EXEC                                                      05957C00 D
                                                                  %                             05958C0C D
1241  1A8C 00C0 000C 0040      OP35F:    OP350 - 1 = MPCR                                      05959C00 D
1242  1A90 00C3 0030 0040               OP351 - 1 = MPCR                                      05960C0C D
1243  1AAC C0C3 0C30 C04C               OP352 - 1 = MPCR                                      05961C00 D
1244  1AB0 00C0 0000 0040               OP353 - 1 = MPCR                                      05962C0C D
                                                                  %                             05963C00 D
                                       OP350:                     % INPUT OUTPUT COMMAND WORD FETCH (5C)  05964C0C D
                                                                  % IOCW IS LOADED PRIOR TO EXECUTING THIS 05965000 D
                                                                  % INSTRUCTION                 05966000 D
1245  4809 2C56 0B30 00FC              LIT AND B = B              % :M: AND :A: FIELDS MUST BE ZERO ELSE  05967C00 D
                                                                  % FAULT ENTERED.              05968C00 D
1246  80FC 0000 0000 0080              15 = LIT; 4 = SAR                                         05969000 D
1247  4809 CC52 000C C0F0              C NEQ B                                                   05970C0C D
1248  600B 00C0 CC0C 00F0              IF TRUE THEN STEP ELSE SKIP                               05971C00 D
1249  3000 00C3 0C30 0040              FAULT - 1 = MPCR                                          05972C00 D
124A  4809 E0C0 B03C C0FC              A3 R = B                                                  05973C00 D
124B  4809 2C56 0B3C C0F0              LIT AND B = B                                             05974C00 D
124C  4809 CC52 0C30 00F0              0 NEQ B                                                   05975C00 D
124D  600B 00C0 CC0C C0FC              IF TRUE THEN STEP ELSE SKIP                               05976C00 D
124E  3C0C 00C0 0030 C040              FAULT - 1 = MPCR                                          05977C00 D
124F  4809 200C 0C1C 00F0              LIT = MAR2                                                05978C0C D
1250  A32C 000C 0C3C C080              IOCW = LIT; 6 = SAR                                       05979C00 D
1251  2F3C 0000 0000 C060              EINPUT - 1 = CPCR          % B = (IOCW)                   05980C00 D
1252  4809 0C40 2000 C0F0              B = A2                     % STORE (IOCW) IN A2            05981C00 D
1253  4809 0C41 0B30 00FC              B L = B                    % ISOLATE NO OF BUFFERS IN UHW OF B  05982C00 D
1254  A000 00C0 0C00 C01C              COMP 18 = SAR                                             05983C00 D
1255  4809 0C40 8B3C 00FC              B R = B                                                   05984C00 D
1256  000C C0C0 000C C010              B = SAR                    % SHIFT NO. OF BUFFERS IN UHW OF B  05985C00 D
1257  4809 EC5C 1030 00F0              A3 OR B = A3               % STORE BUFFER COUNT IN UHW OF A3  05986C0C D
1258  4809 C000 003C C0F0              A2                         % TEST BIT FOR I/O             05987C00 D
1259  580B 00C0 000C C0F0              IF LST THEN STEP ELSE SKIP                                05988000 D
125A  367C 0000 0C30 0040              EOUT - 1 = MPCR                                           05989000 D
125B  3B4C CC03 0030 C040              IN - 1 = MPCR                                             05990C0C D
                                                                  %                             05991C00 D
                                                                  %                             05992C0C D
                                       OP351:                     % RI TYPE 2 BIASED FETCH       05993C00 D
                                                                  % (Y*) BIT 15 INTO CC: 1 = Y* BITS 15,14 05994C30 D
                                                                  % SET CC                      05995C00 D
125C  238C 0000 0C30 0060              CONTENTSRM - 1 = CPCR      % Y* INTO B                    05996C00 D
125D  4809 0C41 0C10 C0FC              B L = BR2                  % Y* INTO BR2                   05997C0C D
```

```
125E  0000 0CC0 0000 0030          COMP 8 = SAR                                  05998C00 D
125F  50E0 00C0 0030 0060          EMULIN - 1 = CPCR      % (Y*) INTO B          05999000 D
1260  48C9 CC41 1830 00F0          B L = B, A3            % (Y*) INTO MS WORD OF B, A3   060CC000 D
1261  0000 0000 0000 0020          COMP 16 = SAR                                 06001000 D
1262  4009 CC40 0C0C 0CF0          B                      % B TO THE ADDER       06002000 D
1263  400E 00C3 000C 00F0          IF NOT MST THEN STEP ELSE SKIP                06003000 D
1264  1AC0 0000 0030 0040          P351 - 1 = MPCR        % POSITIVE CASE        06004000 D
1265  216C 00C0 000C 0060          SET11 - 1 = CPCR       % SET THE CONDITION BITS   06005000 D
1266  4809 00C1 0B3C 00FC          1 L = B                                       06006000 D
1267  200C 0000 0000 0000          COMP 30 = SAR                                 06007000 D
1268  4809 EC5C 8B30 00FC          A3 OR B R = MIR, D     % SET BIT 14 OF (Y*)   06008000 D
1269  0000 00C3 000C C020          16 = SAR                                      06009000 D
126A  4809 0F41 0C10 00FC          BMAR L = BR2           % RESTORE Y* INTO PR2  06010000 D
126B  000C C000 003C C030          COMP 8 = SAR                                  06011000 D
126C  51E0 0000 0C30 C060          EMULOUT - 1 = CPCR                            06012000 D
126D  66EC 00C0 0C0C 0040          OPCODE - 1 = MPCR                             06013000 D
126E  4809 A001 CC3C 40F0  P351:   A1 C = A1,CSAR         % PUT CC BIT 9 INTO LS BIT   06014000 D
126F  1C0C 00C0 3030 0030          25 = SAR                                      06015000 D
1270  4809 A0CF CCC0 C0F0          A1 AND B110 C = A1     % SET BIT 9 TO 0, AND RESTORE A1   06016000 D
1271  48C9 CC41 8B0C C0FB          B C = B                                       06017000 D
1272  A000 00C0 000C 0030          30 = SAR                                      06018000 D
1273  4809 2C5D 8C3C C0FC          LIT OR B C = MIR       % SET (Y*) BITS 14,15  06019000 D
1274  2030 0000 0000 00A0          3 = LIT, 18 = SAR                             06020000 D
1275  4809 0F41 0C10 0CF0          BMAR L = BR2           % RESTORE Y* INTO BR2  06021000 D
1276  000C 00C0 C03C 0030          COMP 8 = SAR                                  06022000 D
1277  51E0 00C0 0C0C C060          EMULOUT - 1 = CPCR                            06023000 D
1278  66EC 00C0 0C30 0040          OPCODE - 1 = MPCR                             06024000 D
                                   %                                             06025000 D
                                   %                                             06026000 D
                   OP352:          % TYPE RK REMOTE EXECUTE                      06027000 D
                                   % EXECUTE (Y), (P) + 2 = P                    06028000 D
1279  4809 2C56 CB3C 00FC          LIT AND B = B          % ISOLATE "M"          06029000 D
127A  00F0 00C0 CC0C C0E0          15 = LIT                                      06030000 D
127B  4809 CC52 0C00 00FC          0 EQL B                                       06031000 D
127C  6819 00C0 0000 C0F0          IF TRUE THEN SKIP                             06032000 D
127D  238C C0C0 0000 0C60          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B        06033000 D
127E  4809 0C41 0B3C C0F0          B L = B                                       06034000 D
127F  0C0C 03C0 0C0C 0020          COMP 16 = SAR                                 06035000 D
1280  4809 EC5C 100C C0F0          A3 OR B = A3           % A3 = (R(M)) OR 0 /INSTRUCTION   06036000 D
1281  633C 00C3 0C30 0060          IFETCH - 1 = CPCR      % "Y" INTO B           06037000 D
1282  4809 E0E0 AC0C C0F0          A3 R = A2              % (R(M)) OR 0 INTO A2   06038000 D
1283  0200 00C0 0030 C0A0          PSW = LIT, 16 = SAR                           06039000 D
1284  4809 CC40 0C30 00F0          A2 + B = MIR           % Y INTO MIR           06040000 D
1285  4809 2003 001C 00F0          LIT = MAR2                                    06041000 D
1286  2F5C C003 0000 C06C          EOUTPUT - 1 = CPCR     % Y INTO PSW           06042000 D
1287  4809 B81C 4C0C 00F0          A1 OR B1C0 = A1        % SET BIT 31 OF SR#1, REMOTE EXEC BIT   06043000 D
1288  66E0 0000 C030 0040          OPCODE - 1 = MPCR                             06044000 D
                                   %                                             06045000 D
                                   %                                             06046000 D
                   OP353:          % TYPE RX BIASED FETCH, (Y) BIT 15 IN CC       06047000 D
                                   % 1 = (Y) BIT 15                              06048000 D
1289  5700 00C0 0C0C C060          RXMFIELD - 1 = CPCR     % Y INTO B            06049000 D
128A  4809 0C41 1C1C C0F0          B L = BR2, A3                                 06050000 D
128B  000C 00C0 0C3C C03C          COMP 8 = SAR                                  06051000 D
128C  60E0 C0C3 0C30 C06C          EMULIN - 1 = CPCR      % (Y) INTO B           06052000 D
128D  4809 0C41 0B3C 00F0          B L = B                                       06053000 D
128E  000C 00C0 003C C020          COMP 16 = SAR                                 06054000 D
128F  4809 CC40 000C 00F0          B                                             06055000 D
1290  400B 00C0 0000 00F0          IF NOT MST THEN SKIP ELSE SKIP                06056000 D
1291  1ADC 00C0 0C0C C040          P353 - 1 = MPCR                               06057000 D
```

```
1292  2160 0000 0000 C060          SET11 - 1 = CPCR      % SET THE CONDITION BITS          06058C00 D
1293  4809 00C1 2000 C0F0          1 L = A2                                                06059C00 D
1294  2000 0000 0C30 0000          COMP 30 = SAR                                           0606CCCC L
1295  4809 CC5C 8B3C C0F0          A2 OR B R = MIR, C    % SET BIT 14 OF (Y)               06061C00 D
1296  000C 000C 0C3C 002C          16 = SAR                                                06062C00 D
1297  4809 E0C3 0010 C0FC          A3 = BR2              % RESTORE Y INTO BR2               06063C0C D
1298  61E0 0000 0C00 C060          EMULOUT - 1 = CPCR                                      06064CC0 D
1299  66E0 00C0 C030 0040          OPCODE - 1 = MPCR                                       06065C00 D
129A  4809 A0C1 CC0C 40F0  P353:   A1 C = A1,CSAR        % PUT CC BIT 9 INTO LS BIT        0606CC0C D
129B  1000 00C3 000C 003C          25 = SAR                                                06067C00 D
129C  4809 A0CF C000 00F0          A1 AND B110 C = A1    % SET BIT 9 TO 0, AND RESTORE A1  06068C00 D
129D  4809 0C41 8B3C 00F0          B C = B                                                 06069C00 D
129E  AC0C 000C 0C30 C030          30 = SAR                                                06C7CC00 D
129F  4809 2C5D 8C3C 00F0          LIT OR D C = MIR      % SET (Y) BIT 14,15               06071C00 D
12A0  2030 C0C3 0000 C0A0          3 = LIT; 18 = SAR                                       06072C00 D
12A1  4809 ECC3 CC1C 00F0          A3 = BR2              % RESTORE Y INTO BR2               06073C00 C
12A2  61E0 0000 000C CC60          EMULOUT - 1 = CPCR                                      06074C0C D
12A3  66E0 0C00 0000 CC40          OPCODE - 1 = MPCR                                       06075C0C D
                                                         %                                 06076C00 D
                                                         %                                 06077C00 D
                                   OPCODE36:             % NOT ASSIGNED. GENERATES INTERRUPT 06078C00 D
                                                         % FAULT.                          06C79CC0 D
12A4  3000 0003 0030 0640          FAULT - 1 = MPCR      % JUMP TO FAULT HANDLING ROUTINE   06C8CCC0 D
                                                         %                                 06081C00 D
                                                         %                                 06082C00 D
                                   OPCODE37:             % NOT IMPLEMENTED                  06083CC0 D
12A5  4E5C 00C0 0C30 0040          NOTIMP - 1 = MPCR                                       06084C0C D
                                                         %                                 06085C00 D
                                   OPCODE40:             % THIS ROUTINE ANALYZES THE 4C OPCODE 06086C00 D
12A6  5F90 0003 0030 0060          XFCODE - 1 = CPCR     % :F: FIELD RETURNED IN A2         06087C00 D
12A7  4809 C0D2 0C3C C0FC          A2 EQL 1                                                06C88C0C D
12A8  6806 00C3 0C0C C0F0          IF TRUE THEN STEP ELSE SKIP                             06C89CC0 D
12A9  1AEC 0CC0 0030 0040          OP401 - 1 = MPCR      % SPECIAL CASE 401                06C9CC0C D
12AA  4809 0C4D 8B0C 0CFC          B R = B               % ISOLATE THE :A: FIELD           06091C00 D
12AB  90FC 0C00 0C00 C080          4 = SAR; 15 = LIT                                       06092CCC D
12AC  4809 2C56 2B30 00FC          LIT AND D = B,A2      % B AND A2 = :A: FIELD            06093C0C D
12AD  4809 2C5E 3000 00F0          LIT GEQ B                                               06094C00 D
12AE  00C0 0000 0C0C C0E0          12 = LIT              % TEST FOR NOT ASSIGNED INST.     06095CC0 D
12AF  7819 0003 0030 C0F0          IF TRUE THEN SKIP                                       06056CC0 D
1280  3C0C C003 0C3C 0040          FAULT - 1 = MPCR                                        06097C00 D
1281  4809 C640 0C4C 00F0          A2 + AMPCR = AMPCR                                      06098C00 D
1282  1AFC 00C3 0000 C0C0          JUMP40 - 1 = AMPCR                                      06099C00 D
1283  4809 0C00 0030 00F0          STEP                                                    0610CC00 D
1284  4824 0000 0C30 00F0          EXEC                                                    0610DC00 D
                                                         %                                 0610E000 D
1285  1B0C 0CC0 0C30 0040  JUMP4C: OP40X00 - 1 = MPCR                                      0610C0C0 D
1286  1B1C 0000 0030 C040          OP40X01 - 1 = MPCR                                      06104C0C D
1287  1B2C 00C0 0000 C040          OP40X02 - 1 = MPCR                                      0610C000 D
1288  1B30 0000 00C0 0040          OP40X03 - 1 = MPCR                                      0610C000 D
1289  1B40 0003 0C00 C040          OP40X04 - 1 = MPCR                                      0610C0C0 D
128A  1B5C 00C0 000C C04C          OP40X05 - 1 = MPCR                                      06108C00 D
128B  4E5C 00C0 0C3C 0040          NOTIMP - 1 = MPCR                                       0610C000 D
128C  4E5C 00C0 C030 C040          NOTIMP - 1 = MPCR                                       06110C00 D
128D  186C 00C3 00C0 0040          OP40X10 - 1 = MPCR                                      06111C00 D
128E  187C 0CC3 003C C040          OP40X11 - 1 = MPCR                                      06112C00 D
128F  1880 0C00 003C C040          OP40X12 - 1 = MPCR                                      06113CC0 D
12C0  1B90 0000 0000 C040          OP40X13 - 1 = MPCR                                      06114C0C D
                                                         %                                 06115C00 D
                                   OP4GX00:              % CC INDICATES = OR 0; JUMP EQUAL  06116C0C D
12C1  226C 0C00 0000 6060          CHECKCC - 1 = CPCR    % CC RETURNED IN B                06117C3C D
```

```
12C2  4809 0C52 0000 00F0          0 EQL B                                          06118C00 D
12C3  63C9 0000 0000 00F0          IF FALSE THEN SET LC1                            06119C00 D
12C4  18AC 0000 003C 0040          OP40X - 1 = MPCR                                 06120C00 D
                                                                                    06121C00 D
                                   OP40X01:                 % CC NOT EQUAL OR NOT 0; JUMP NOT EQUAL 06122C00 D
12C5  2260 00C0 0030 0060          CHECKCC - 1 = CPCR                               06123C00 D
12C6  4809 0C5E 0000 00F0          0 GTR 0                                          06124C00 D
12C7  78C9 0000 0030 C0FC          IF FALSE THEN SET LC1                            06125C00 D
12C8  18DC 0000 0030 0040          OP40X - 1 = MPCR                                 06126C00 D
                                                                                    06127C00 D
                                   OP40X02:                 % CC >= OR + ; JUMP GTR OR EQUAL 06128C00 D
12C9  2260 00C0 000C 0060          CHECKCC - 1 = CPCR                               06129C00 D
12CA  4809 2C5E 0030 00F0          LIT GEQ B                                        06130C00 D
12CB  0C2C 0000 0030 00E0          2 = LIT                                          06131C00 D
12CC  73C9 0000 0030 C0F0          IF FALSE THEN SET LC1                            06132C00 D
12CD  1DCC 0000 000C 0040          OP40X - 1 = MPCR                                 06133C00 D
                                                                                    06134C00 D
                                   OP40X03:                 % CC < OR -; JUMP LESS  06135C00 D
12CE  2260 00C0 0030 0060          CHECKCC - 1 = CPCR                               06136C00 D
12CF  4809 2C52 0030 00F0          LIT EQL E                                        06137C00 D
12D0  0C3C 0000 0030 00E0          3 = LIT                                          06138C00 D
12D1  53C9 00C0 0030 C0F0          IF FALSE THEN SET LC1                            06139C00 D
12D2  18D0 0000 000C C040          OP40X - 1 = MPCR                                 06140C00 D
                                                                                    06141C00 D
                                   OP40X04:                 % IF OVERFLOW SET; JUMP; JUMP OVERFLOW 06142C00 D
12D3  4809 A0C0 A030 00F0          A1 R = A2                                        06143C00 D
12D4  3000 0000 0000 0C3C          27 = SAR                                         06144C00 D
12D5  4809 C000 0030 00F0          A2                                               06145C00 D
12D6  53C9 00C0 0C00 C0F0          IF NOT LST THEN SET LC1                          06146C00 D
12D7  18E0 00C0 C03C 0040          OP40X - 1 = MPCR                                 06147C00 D
                                                                                    06148C00 D
                                   OP40X05:                 % IF CARRY; JUMP CARRY  06149C00 D
12D8  4009 A0C0 A020 00F0          A1 R = A2                                        06150C00 D
12D9  8000 00C0 003C C030          28 = SAR                                         06151C00 D
12DA  4809 C0C0 003C 00F0          A2                                               06152C00 D
12DB  53C9 0000 0030 00FF          IF NOT LST THEN SET LC1                          06153C00 D
12DC  18F0 00C0 0000 C040          OP40X - 1 = MPCR                                 06154C00 D
                                                                                    06155C00 D
                                   OP40X10:                 % JUMP                  06156C00 D
12DD  1C00 0000 0000 C04C          OP40X - 1 = MPCR                                 06157C00 D
                                                                                    06158C00 D
                                   OP40X11:                 % JUMP AFTER STOP       06159C00 D
12DE  48CC C0C0 0030 60F0          WAIT                     % MACHINE STOPS         06160C00 D
12DF  4809 00C0 000C 00F0          STEP                                             06161C00 D
12E0  1C10 00C0 003C 0040          OP40X - 1 = MPCR                                 06162C00 D
                                                                                    06163C00 D
                                   OP40X12:                 % IF KEY 1 SET, STOP. JUMP. 06164C00 D
12E1  4809 A0C0 A030 C0F0          A1 R = A2                                        06165C00 D
12E2  9000 0000 0C0C C03C          29 = SAR                                         06166C00 D
12E3  4809 C0C0 C030 00F0          A2                                               06167C00 D
12E4  53CB 00C0 0C00 00F0          IF NOT LST THEN SET LC1 ELSE SKIP                06168C00 D
12E5  1C20 00C0 0C00 0040          OP40X - 1 = MPCR                                 06169C00 D
12E6  48C0 00C0 0C30 C0F0          WAIT                     % MACHINE STOPS         06170C00 D
12E7  4809 00C0 00CC 0CF0          STEP                                             06171C00 D
12E8  1C30 00C0 003C 004C          OP40X - 1 = MPCR                                 06172C00 D
                                                                                    06173C00 D
                                   OP40X13:                 % IF KEY 2 SET, STOP. JUMP 06174C00 D
12E9  4809 A0C0 A0C0 C0F0          A1 R = A2                                        06175C00 D
12EA  A00C 00C0 0000 C030          30 = SAR                                         06176C00 D
12EB  4809 C0C0 0030 00F0          A2                                               06177C00 D
```

```
12EC  53C8 0000 0000 00F0          IF NOT LST THEN SET LC1 ELSE SKIP                    06178000 D
12ED  5819 0000 0000 00F0          IF LST THEN SKIP                                     06179000 D
12EE  1040 0000 0000 0040          OP40X - 1 = MPCR                                     06180000 D
12EF  4800 00ND 0000 00F0          WAIT                        % MACHINE STOPS          06181000 D
12F0  4809 0000 0000 00F0          STEP                                                 06182000 D
12F1  1050 0000 0000 0040          OP40X - 1 = MPCR                                     06183000 D
                                                               %                        06184000 D
12F2  5F90 0000 0000 0060    OP40X:  XFCODE - 1 = CPCR         % DECODE THE :F: FIELD    06185000 D
12F3  4809 C640 0000 00FC          A2 + AMPCR = AMPCR                                   06186000 D
12F4  1060 0000 0000 00C0          OP40F - 1 = AMPCR                                    06187000 D
12F5  4809 0000 0000 00F0          STEP                                                 06188000 D
12F6  4824 0000 0030 00F0          EXEC                                                 06189000 D
                                                               %                        06190000 D
12F7  1070 0000 0000 0040    OP40F:  OP400 - 1 = MPCR                                   06191000 D
12F8  3000 0000 0000 0040          FAULT - 1 = MPCR            % DECODING PROBLEM       06192000 D
12F9  1080 0000 0000 0040          OP402 - 1 = MPCR                                     06193000 D
12FA  1090 0000 0000 0040          OP403 - 1 = MPCR                                     06194000 D
                                                               %                        06195000 D
                                   OP400:                      % THIS ROUTINE LOADS (R(M)) INTO P.  06196000 D
                                                               % TYPE RR                06197000 D
12FB  280P 0000 0000 00F0          IF LC1 THEN STEP ELSE SKIP  % ABORT JUMP, GET NEXT INST. 06198000 D
12FC  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                    06199000 D
12FD  4809 E155 0B00 00F0          A3 AND LIT = B              % B = :M: FIELD          06200000 D
12FE  00F0 0000 0000 00EC          15 = LIT                                             06201000 D
12FF  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                                  06202000 D
1300  2F30 0000 0000 0060          EINPUT - 1 = CPCR           % B = (R(M))             06203000 D
1301  4809 A0D3 0000 00F0          A1 R = A1                                            06204000 D
1302  0000 0000 0000 0020          16 = SAR                                             06205000 D
1303  4809 A0E1 0000 00F0          A1 L = A1                                            06206000 D
1304  4809 AC50 0000 00F0          A1 OR B = A1                % (R(M)) -> P             06207000 D
1305  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                    06208000 D
                                                               %                        06209000 D
                                   OP402:                      % THIS ROUTINE LOADS Y INTO P. TYPE RX  06210000 D
1306  280B 0000 0000 00F0          IF LC1 THEN STEP ELSE SKIP  % ABORT JUMP, GET NEXT INST. 06211000 D
1307  1D30 0000 0000 0040          BUMP - 1 = MPCR                                      06212000 D
1308  4839 E155 0B00 00F0          A3 AND LIT = B                                       06213000 D
1309  00FF 0000 0000 00E0          15 = LIT                                             06214000 D
130A  4809 0052 0000 00F0          0 EQL B                                              06215000 D
130B  6A49 0000 0000 00FC          IF TRUE THEN SET LC2                                 06216000 D
130C  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                                  06217000 D
130D  2F30 0000 0000 0060          EINPUT - 1 = CPCR           % B = (R(M))             06218000 D
130E  4809 0C40 1000 00F0          B = A3                      % A3 = (R(M))            06219000 D
130F  633C 0000 0000 0060          IFETCH - 1 = CPCR           % B = :Y:                06220000 D
1310  3819 0000 0000 00FC          IF LC2 THEN SKIP                                     06221000 D
1311  280B 0000 0000 00F0          IF LC1 THEN STEP ELSE SKIP  % ABORT JUMP, GET NEXT INST. 06222000 D
1312  1D30 0000 0000 0040          BUMP - 1 = MPCR                                      06223000 D
1313  4809 EC40 0B00 00FC          A3 + B = B                  % (R(M)) + :Y: = Y       06224000 D
1314  4809 A0D3 0000 00F0          A1 R = A1                                            06225000 D
1315  0000 0000 0000 0020          16 = SAR                                             06226000 D
1316  4809 A0E1 0000 00F0          A1 L = A1                                            06227000 D
1317  4809 AC50 0000 00F0          A1 OR B = A1                % Y INTO PAR              06228000 D
1318  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                    06229000 D
                                                               %                        06230000 D
                                   OP403:                      % THIS ROUTINE LOADS Y INTO P.  06231000 D
                                                               % TYPE RX.               06232000 D
1319  5700 0000 0000 0060          RXMFIELD - 1 = CPCR         % B = Y                   06233000 D
131A  4809 CC41 0010 00F0          B L = BR2                                            06234000 D
131B  0000 00C0 0000 0030          COMP B = SAR                                         06235000 D
131C  50E0 0000 0000 0060          EMULIN - 1 = CPCR           % B = (Y)                06236000 D
131D  4809 A0D3 0000 00FC          A1 R = A1                                            06237000 D
```

```
131E  0000 0000 0000 0020          16 = SAR                                          06238000 D
131F  4809 AC01 4090 60F0          A1 L = A1                                         06239000 D
1320  4809 AC5C 4090 00F0          A1 OR D = A1          % (Y) INTO PAR              06240000 D
1321  66E0 0000 0090 0040          OPCODE - 1 = MPCR                                 06241000 D
                                                       %                           06242000 D
                                   OP401:              % LJ: LOCAL JUMP             06243000 D
                                                       % (P) + XD -> P             06244000 D
                                                       % TYPE RI (1)               06245000 D
1322  559C 0000 0090 004C          RI1 - 1 = MPCR        % (P) + XD -> P            06246000 D
                                                       %                           06247000 D
                                                       %                           06248000 D
                                   OPCODE41:           % INDEX JUMP                 06249000 D
1323  5F90 0000 0030 0060          XFCODE - 1 = CPCR     % "F" CODE RETURNED IN A2    06250000 D
1324  4809 C640 009C C0FC          A2 + AMPCR = AMPCR                                06251000 D
1325  1CAC 0003 0030 00C0          OP41F - 1 = AMPCR                                 06252000 D
1326  4009 0000 C60C C0F0          STEP                                              06253000 D
1327  4824 0003 0000 00FC          EXEC                                              06254000 D
                                                       %                           06255000 D
                                                       %                           06256000 D
1328  1CB0 0000 009C 0040          OP41F:  OP410 - 1 = MPCR                          06257000 D
1329  1CCC 0000 0090 0040          OP411 - 1 = MPCR                                  06258000 D
132A  1CDC 0000 003C 004C          OP412 - 1 = MPCR                                  06259000 D
132B  1CEC 0000 000C 0040          OP413 - 1 = MPCR                                  06260000 D
                                                       %                           06261000 D
                                                       %                           06262000 D
                                   OP410:              % RR TYPE INDEX JUMP         06263000 D
                                                       % IF (R(A)) NE 0, (R(A)) - 1 INTO R(A)  06264000 D
                                                       % AND (R(M)) IN P            06265000 D
132C  2260 0000 0030 0060          CONTENTSRA - 1 = CPCR  % RETURN (R(A)) IN B       06266000 D
132D  4809 0052 0030 00F0          D EQL 0                                           06267000 D
132E  68DB 0000 0090 00F0          IF TRUE THEN STEP ELSE SKIP                       06268000 D
132F  66E0 0003 0090 0040          OPCODE - 1 = MPCR                                 06269000 D
1330  4809 0C40 209C 00F0          B = A2                                            06270000 D
1331  4809 C0DE 003C C0F0          A2 - 1 = MIR          % DECREMENT (R(A))          06271000 D
1332  2F50 0000 0000 0060          EOUTPUT - 1  = CPCR   % WRITE NEW (R(A)) VALUE IN R(A)  06272000 D
1333  2380 0000 0000 0060          CONTENTSRM - 1 = CPCR  % B CONTAINS (R(M))        06273000 D
1334  4809 AC00 003C 00F0          A1 R = A1             % CLEAR THE OLD PAR          06274000 D
1335  0000 0000 0000 0020          16 = SAR                                          06275000 D
1336  4809 AC01 4000 00F0          A1 L = A1                                         06276000 D
1337  4809 AC5C 409C 00F0          A1 OR B = A1          % CREATE NEW PAR             06277000 D
1338  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                 06278000 D
                                                       %                           06279000 D
                                                       %                           06280000 D
                                   OP411:              % RI TYPE 1, LOCAL JUMP INDIRECT  06281000 D
                                                       % IF (CC) NE 0, ((P) + D) INTO P  06282000 D
1339  2260 0000 0090 0060          CHECKCC - 1 = CPCR    % PUT (CC) INTO LS 2 BITS OF B  06283000 D
133A  4809 0052 009C C0FC          B EQL 0                                           06284000 D
133B  68DB 0000 0090 60F0          IF TRUE THEN STEP ELSE SKIP                       06285000 D
133C  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                 06286000 D
133D  5590 0000 0000 004C          RI1 - 1 = MPCR        % (P) + D = P               06287000 D
                                                       %                           06288000 D
                                                       %                           06289000 D
                                   OP412:              % PK TYPE INDEX JUMP         06290000 D
                                                       % IF (R(A)) NE 0, (R(A)) - 1 INTO R(A),  06291000 D
                                                       % Y INTO P                   06292000 D
133E  2290 0003 0030 0060          CONTENTSRA - 1 = CPCR  % PUT (R(A)) IN B          06293000 D
133F  4809 0052 0000 00FC          B EQL 0                                           06294000 D
1340  68DB 0000 009C C0FC          IF TRUE THEN STEP ELSE SKIP                       06295000 D
1341  66E0 0003 0030 0040          OPCODE - 1 = MPCR                                 06296000 D
1342  4809 0C40 209C 00FC          B = A2                                            06297000 D
```

```
1343  4809 C0DE C050 C0F0          A2 - 1 = MIR              % DECREMENT (R(A))                  06298C00 D
1344  4809 E0C3 801C 00F0          A3 R = MAR2               % ADDRESS OF R(A) INTO MAR2          06299C00 D
1345  0000 0000 0030 0020          16 = SAR                                                      06300000 D
1346  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR        % (R(A)) - 1  INTO R(A)             06301000 D
1347  4809 E155 1000 C0F0          A3 AND LIT = A3           % "M" FIELD IN A3                   06302000 D
1348  00FC 00C0 0000 00E0          15 = LIT                                                      06303000 D
1349  4809 E012 0030 00F0          A3 EQL 0                                                      06304000 D
134A  6C19 C000 0000 00F0          IF TRUE THEN 0 = D; SKIP                                      06305000 D
1340  2380 00C3 C030 006C          CONTENTSRM - 1 = CPCR     % RETURN (R(M)) IN D                06306000 D
134C  4809 0C40 1000 00FC          B = A3                    % STORE (R(M)) OR C IN A3           06307000 D
134D  6330 0000 0000 006C          IFETCH - 1 = CPCR         % PUT Y IN B                        06308000 D
134E  4809 EC40 0B0C 00F0          A3 + B = B                % Y INTO B                           06309000 D
134F  4809 A000 C00C C0F0          A1 R = A1                                                     06310000 D
1350  0000 00C0 0030 0020          COMP 16 = SAR                                                 06311000 D
1351  4809 A0C1 4C3C C0FC          A1 L = A1                                                     06312000 D
1352  4809 AC5C 4000 00F0          A1 OR D = A1                                                  06313000 D
1353  56EC 0000 0000 0040          OPCODE - 1 = MPCR                                             06314000 D
                                                       %                                        06315000 D
                                                       %                                        06316000 D
                                   OP413:              % RX TYPE INDEX JUMP                      06317000 D
                                                       % IF (R(A)) NE 0, (R(A)) - 1 = (R(A)),   06318000 D
                                                       % (Y) INTO P                             06319000 D
1354  4809 E000 8B3C 00FC          A3 R = B                  % SHIFT A3 TO OBTAIN "A"             06320000 D
1355  80F0 00C3 0000 008C          4 = SAR; 15 = LIT                                             06321000 D
1356  4809 2C56 0B3C 00F0          LIT AND B = B             % ISOLATE "A"                       06322000 D
1357  51CC 00C0 0030 0060          REGSTACK - 1 = CPCR       % R(A) INTO MAR2                    06323000 D
1358  2F3C 00C0 0030 C060          EINPUT - 1 = CPCR         % (R(A)) INTO B                     06324000 D
1359  4809 0CE2 0C3C C0F0          B EQL C                                                       06325000 D
135A  58C0 C000 00C0 00F0          IF TRUE THEN STEP ELSE SKIP                                   06326000 D
135B  1030 00C0 0000 0040          BUMP - 1 = MPCR           % ADVANCE THE PAR BY 1 AND CALL OPCODE 06327000 D
135C  4809 0C40 200C 00FC          B = A2                                                        06328000 D
135D  4809 C0DE C030 00FC          A2 - 1 = MIR              % DECREMENT (R(A))                  06329000 D
135E  2F5C 0000 0C0C C060          EOUTPUT - 1 = CPCR        % OUTPUT NEW VALUE OF R(A) IN R(A)  06330000 D
135F  4809 EC00 0B0C 00F0          A3 = B                    % RESTORE INSTRUCTION IN B           06331000 D
1360  5700 00C0 0030 0060          RXMFIELD - 1 = CPCR       % ANALYZE THE "M" FIELD             06332000 D
1361  4809 0C41 0010 00FC          B L = BR2                 % ADDRESS OF Y IN BR2                06333000 D
1362  0C0C C0C3 0C3C C03C          COMP 8 = SAR                                                  06334000 D
1363  60EC C0C0 0000 C060          EMULIN - 1 = CPCF         % (Y) INTO B                        06335000 D
1364  4809 A0C0 C030 00F0          A1 R = A1                 % CLEAR OLD PAR                      06336000 D
1365  000C 00C0 0000 0020          COMP 16 = SAR                                                 06337000 D
1366  4809 A0C1 4C3C C0F0          A1 L = A1                                                     06338000 D
1367  4809 AC5C 4C30 0CFC          A1 OR B = A1              % CREATE NEW PAR                     06339000 D
1368  66E0 00C0 C03C C040          OPCODE - 1 = MPCR                                             06340000 D
                                                       %                                        06341000 D
                                                       %                                        06342000 D
                                   OPCODE42:           % RR TYPE JUMP, LINK REGISTERS            06343000 D
1369  5F90 00C0 0C3C C060          XFCODE - 1 = CPCR         % "F" INTO A2                       06344000 D
136A  4809 C640 0C4C 00F0          A2 + AMPCR = AMPCR                                            06345000 D
136B  1CF0 00C3 0000 00C0          OP42F - 1 = AMPCR                                             06346000 D
136C  4809 C0C0 0C3C 00FC          STEP                                                          06347000 D
136D  4824 C0C0 C030 00F0          EXEC                                                          06348000 D
                                                       %                                        06349000 D
                                                       %                                        06350000 D
136E  100C 00C0 000C 004C          OP42F:  OP42C - 1 = MPCR                                      06351000 D
136F  3000 0000 0C3C C04C          FAULT - 1 = MPCR                                              06352000 D
1370  101C 0000 0000 C04C          OP422 - 1 = MPCR                                              06353000 D
1371  1020 00C3 0C3C C040          OP423 - 1 = MPCR                                              06354000 D
                                                       %                                        06355000 D
                                                       %                                        06356000 D
                                   OP420:              % JUMP, LINK REGISTER                     06357000 D
```

```
1372  4809 0C40 8B0C C0F0          B R = B                                  % (P) + 1 INTO R(A)); (R(M)) INTO P      06358C0C D
1373  30FC 00C0 C00C 008C          4 = SAR; 15 = LIT                                                                 06359C0C D
1374  4809 2C56 0B0C C0FC          LIT AND B = B                            % B CONTAINS "A"                         06360C0C D
1375  51C0 00C0 0C00 006C          REGSTACK - 1 = CPCR                      % ADDRESS OF R(A) IN MAR2                06361C0C D
1376  4809 A001 2C00 C0F0          A1 L = A2                                % (P) INTO MS WORD OF A2                  06362C0C D
1377  000C 0000 0000 0020          16 = SAR                                                                          06363C0C D
1378  4809 C0C0 A00C 00FC          A2 R = A2                                % (P) INTO LS WORD OF A2                  06364C0C D
1379  4009 C0C0 003C 0010          A2 + 1 = MIR                             % (P) + 1 INTO MIR                       06365CC0 D
137A  2F5C 00C0 0C00 0060          EOUTPUT - 1 = CPCR                       % (P) + 1 INTO R(A)                      06366C0C D
1378  2380 C0C3 C00C 006C          CONTENTSRM - 1 = CPCR                    % (R(M)) INTO B                          06367C0C D
137C  4009 AC00 CC3C C0F0          A1 R = A1                                % CLEAR PAR FIELD                         06368C0C D
137D  000C C0C0 0C00 C020          COMP 16 = SAR                                                                     06369C0C D
137E  4809 A001 4C0C C0F0          A1 L = A1                                % PREPARE FOR NEW PAR                     0637CC0C D
137F  4809 AC5C 400C 00F0          A1 OR B = A1                             % CREATE NEW PAR                          0637AC0C D
1380  66EC 00C0 0C00 C040          OPCODE - 1 = MPCR                                                                 0637BC0C D
                                                                            %                                        0637CC0C D
                                                                            %                                        0637DC0C D
                                   OP422:                                   % RK TYPE JUMP, LINK REGISTERS            0637EC0C D
                                                                            % (P) + 2 INTO R(A)); Y INTO P            0637FC0C D
1381  4809 0C40 8B0C 00FC          B R = B                                                                           06380C0C D
1382  80F0 0FC3 00FC C08C          4 = SAR; 15 = LIT                                                                 06381C0C D
1383  4809 2C56 0B0C C0FC          LIT AND B = B                            % B CONTAINS "A"                         06382C0C D
1384  51C0 C0C0 0C00 0060          REGSTACK - 1 = CPCR                      % MAR2 CONTAINS ADDRESS OF R(A)          06383C0C D
1385  4809 A001 203C 00F0          A1 L = A2                                % SAVE OLD PAR                            06384C0C D
1386  0020 0000 0C3C C0A0          COMP 16 = SAR; 2 = LIT                                                            06385C0C D
1387  4809 C0C0 AC0C 00F0          A2 R = A2                                % PAR INTO LS WORD OF A2                  06386C0C D
1388  4809 C140 0C30 C0F0          A2 + LIT = MIR                           % MIR CONTAINS (P) + 2                   06387C0C D
1389  2F5C C0C0 0C00 006C          EOUTPUT - 1 = CPCR                       % PUT (P) + 2 INTO R(A)                  06388C0C D
138A  4809 E156 103C 00FC          A3 AND LIT = A3                          % "M" INTO A3                            06389C0C D
138B  00FC 0FC0 0C3C C0E0          15 = LIT                                                                          0638AC0C D
138C  4809 E012 0C00 C0F0          A3 EQL 0                                                                          0638BC0C D
138D  6C19 0000 0B0C 00FC          IF TRUE THEN 0 = 0; SKIP  % CHECK IF "M" = 0                                      0638CC0C D
138E  2380 0000 C030 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B                                            0638DC0C D
138F  4809 0C40 100C 00F0          B = A3                                                                            0638EC0C D
1390  633C 00C0 000C 0C60          IFETCH - 1 = CPCR                        % Y INTO B                               0638FC0C D
1391  4809 EC40 CB3C 00F0          A3 + B = B                               % (Y) INTO B                             06390C0C D
1392  4809 A0C3 C030 C0F0          A1 R = A1                                % CLEAR THE PAR                          06391C0C D
1393  0000 00C0 0C00 C020          COMP 16 = SAR                                                                     06392C0C D
1394  4809 A001 400C C0F0          A1 L = A1                                % PREPARE FOR THE NEW PAR                 06393C0C D
1395  4809 AC5C 4CC0 00F0          A1 OR D = A1                             % CREATE THE NEW PAR                      06394C0C D
1396  56EC 0000 0C00 C040          OPCODE - 1 = MPCR                                                                 06395C0C D
                                                                            %                                        06396C0C D
                                                                            %                                        06397C0C D
                                   OP423:                                   % RX TYPE JUMP, LINK REGISTERS            06398C0C D
                                                                            % (P) + 2 INTO R(A)); (Y) INTO P          06399C0C D
1397  4809 0C40 9B0C 00FC          B R = B                                                                           0639AC0C D
1398  80FC 0000 0C30 0080          4 = SAR; 15 = LIT                                                                 0639BC0C D
1399  4809 2C56 0B30 C0FC          LIT AND B = F                            % B CONTAINS "A"                         0639CC0C D
139A  51C0 00C3 0030 C060          REGSTACK - 1 = CPCR                      % ADDRESS OF R(A) IN MAR2                0639DC0C D
1398  4809 A001 20CC C0FC          A1 L = A2                                % (P) INTO MS WORD OF A2                  0639EC0C D
139C  0C2C C000 CC3C 00A0          16 = SAR; 2 = LIT                                                                 0639FC0C D
139D  4809 C0C0 A0CC 00F0          A2 R = A2                                % (P) INTO LS WORD OF A2                  063A0C0C D
139E  4809 C140 0030 C0F0          A2 + LIT = MIR                           % (P) + 2 INTO MIR                       063A1C0C D
139F  2F5C C0C0 0C3C C060          EOUTPUT - 1 = CPCR                       % (P) + 2 INTO R(A)                      063A2C0C D
13A0  4809 E0C0 0B0C 00F0          A3 = B                                   % SET UP D FOR RXMFIELD                  063A3C0C D
13A1  57CC 0000 00CC 0060          RXMFIELD - 1 = CPCR                      % Y INTO B                               063A4C0C D
13A2  4809 0C41 0C10 00FC          B L = BR2                                % Y INTO BR2                              063A5C0C D
13A3  00CC C0C0 000C C030          COMP B = SAR                                                                      063A6C0C D
13A4  60E0 00C0 0C00 C060          EMULIN - 1 = CPCR                        % (Y) INTO B                             063A7C0C D
```

```
13A5  4809 ACC0 CC00 00F0          A1 R = A1                    % CLEAR THE OLD PAR                  06418C00 D
13A6  0000 C003 0000 0020          COMP 16 = SAR                                                     06419C00 D
13A7  4809 AC01 4030 C0F0          A1 L = A1                    % PREPARE TO RECEIVE NEW PAR         06420000 D
13A8  4809 AC5C 4030 C0F0          A1 OR B = A1                 % CREATE NEW PAR                      06421C00 D
13A9  66EC 0003 C090 C040          OPCODE - 1 = MPCR                                                 06422C00 D
                                                                %                                    06423C00 D
                                                                %                                    06424C00 D
                                   OPCODE43:                    % JUMP, LINK MEMORY                   06425C00 D
13AA  5F90 CCC0 0C00 0060          XFCODE - 1 = CPCR            % "F" RETURNED IN A2                  06426CC0 D
13AB  4809 C640 CC40 C0F0          A2 + AMPCR = AMPCR                                                06427C00 D
13AC  1D3C 0003 0000 00C0          OP43F - 1 = AMPCR                                                 06428C00 D
13AD  4809 00C0 0C3C C0F0          STEP                                                              06429C00 D
13AE  4824 0C00 0C00 00FC          EXEC                                                              06430CC0 D
                                                                %                                    06431CC0 D
                                                                %                                    06432C00 D
13AF  3000 0000 0070 0040          OP43F:  FAULT - 1 = MPCR                                          06433C00 D
13B0  1D40 00C3 003C C040          OP431 - 1 = MPCR                                                  06434CCC D
13B1  1D5C 0C00 0C00 C040          OP432 - 1 = MPCR                                                  06435C06 D
13B2  1D60 CCC0 0000 0040          OP433 - 1 = MPCR                                                  06436C00 D
                                                                %                                    06437C00 D
                                                                %                                    06438C00 D
                                   OP431:                       % RI TYPE 1, LOCAL JUMP, LINK MEMORY  06439C00 D
                                                                % ONLY EXECUTE IF CC = 0              06440C00 D
13B3  2260 00C0 CC0C 0060          CHECKCC - 1 = CPCR           % CHECK THE CC CODE                   06441C00 D
13B4  4809 0C52 0C00 C0F0          0 EQL B                      % CHECK IF CC = 0                     06442C00 D
13B5  6819 0000 0C00 00F0          IF TRUE THEN SKIP                                                 06443C00 D
13B6  66E0 00C3 0000 0040          OPCODE - 1 = MPCR                                                 06444C00 D
13B7  2809 A001 2000 00F0          A1 L = A2; IF LC1            % PAR INTO A2 (MS WORD)               06445C00 D
13B8  00CC C003 0000 0020          COMP 16 = SAR                                                     06446C00 D
13B9  4809 C0C0 A000 00FC          A2 R = A2                    % PAR INTO A2 (LS WORD)               06447C00 D
13BA  4809 C0C0 003C 00F0          A2 + 1 = MIR                 % (P) + 1 INTO MIR                    06448C00 D
13BB  4809 C0C1 2C00 C0FC          A2 L = A2                    % PAR INTO UNW                        06449C00 D
13BC  4809 E000 CB30 00F0          A3 = B                       % RESTORE INSTRUCTION INTO B          06450C00 D
13BD  4809 CC41 0B30 40F0          B L = B;CSAR                 % "D" SIGN BIT IN MS BIT OF B          06451C00 D
13BE  0000 0003 C00C C010          COMP 24 = SAR                                                     06452C00 D
13BF  4809 0C40 0000 C0FC          B                                                                 06453CC0 D
13C0  4FC9 CC5E CB3C 00FC          IF MST THEN - B = B; SET LC1  % IF "D" NEGATIVE SET LC1           06454C00 D
13C1  4809 0C40 8B0C C0FC          B R = B                      % "D" MAGNITUDE IN LSB OF B           06455C00 D
13C2  4809 2C57 2B30 C0F0          LIT AND B L = B              % "D" MAGNITUDE IN B                  06456C0C D
13C3  07FC 0000 0C3C 00A0          127 = LIT; COMP 16 = SAR                                          06457CCC D
13C4  4809 C000 0C3C 00F0          STEP                                                              06458C06 D
13C5  0000 00C3 0C3C C01C          B = SAR                                                           06459C00 D
13C6  2C19 CC5E 9010 C0F0          IF LC1 THEN A2 - B R = BR2,A3; SKIP  % (P) - D                    06460C00 D
13C7  4809 CC40 9C10 C0FC          A2 + B R = BR2,A3            % (P) + D                            06461CC0 D
13C8  61E0 00C3 003C 0060          EMULOUT - 1 = CPCR           % (P) + 1 INTO (P) + D               06462C0C D
13C9  4809 E0C0 8B00 C0F0          A3 R = B                                                          06463C00 D
13CA  00C0 00C0 C030 C010          B = SAR                                                           06464C00 D
13CB  4809 CC46 CB3C 00F0          B + 1 = B                    % (P) + D + 1 INTO B                 06465C00 D
13CC  4809 A0C0 CC00 00F0          A1 R = A1                    % CLEAR OLD PAR                       06466C00 D
13CD  0000 0000 0070 0020          COMP 16 = SAR                                                     06467C00 D
13CE  4809 AC01 4C0C 00F0          A1 L = A1                                                         06468C00 D
13CF  4809 AC5C 4C3C 00F0          A1 OR B = A1                 % CREATE NEW PAR                      06469C00 D
13D0  66E6 0000 0000 0040          OPCODE - 1 = MPCR                                                 06470C00 D
                                                                %                                    06471C00 D
                                                                %                                    06472C00 D
                                   OP432:                       % RK TYPE JUMP, LINK MEMORY           06473C00 D
                                                                % (P) + 2 INTO Y; Y + 1 INTO P        06474CC0 D
13D1  4809 2C56 0B00 00F0          LIT AND B = B                % ISOLATE "N" FIELD                   06475C0C D
13D2  00FC 00C0 0030 00E0          15 = LIT                                                          06476C00 D
13D3  4809 0C52 0C30 00F0          B EQL 0                      % IS "N" = 0 ?                       06477C00 D
```

```
13D4  6819 0000 0000 00F0          IF TRUE THEN SKIP                                  06478000 D
13D5  2380 00C0 0030 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO B              06479000 D
13D6  4809 0C40 1000 00F0          D = A3                                             06480000 D
13D7  533C 0000 C030 0060          IFETCH - 1 = CPCR      % Y INTO B                  06481000 D
13D8  4809 EC41 1010 00F0          A3 + B L = BR2,A3      % Y INTO BR2, A3             06482000 D
13D9  0000 0000 0000 C030          COMP 8 = SAR                                       06483000 D
13DA  48C9 A001 2030 00F0          A1 L = A2              % (P) INTO A2               06484000 D
13DB  002C 00C0 C030 00A0          COMP 16 = SAR; 2 = LIT                             06485000 D
13DC  4809 C0C0 A000 00F0          A2 R = A2              % (P) INTO LS WORD OF A2     06486000 D
13DD  4809 C140 0030 00FC          A2 + LIT = MIR         % MIR CONTAINS (P) + 2       06487000 D
13DE  61E0 0000 C030 0060          EMULOUT - 1 = CPCR     % (P) + 2 INTO Y             06488000 D
13DF  4809 E0C0 9000 00F0          A3 R = A3              % Y INTO A3 LS WORD          06489000 D
13E0  0000 0000 0000 C010          8 = SAR                                            06490000 D
13E1  48C9 E0C0 0B30 00F0          A3 + 1 = B             % B CONTAINS Y + 1           06491000 D
13E2  4809 AC00 C00F 00F0          A1 R = A1              % CLEAR THE PAR              06492000 D
13E3  0000 00C0 0000 0020          16 = SAR                                           06493000 D
13E4  4809 A001 4000 00F0          A1 L = A1                                          06494000 D
13E5  4809 AC5C 4070 00F0          A1 OR B = A1           % CREATE NEW PAR             06495000 D
13E6  66E0 00C0 0000 C040          OPCODE - 1 = MPCR                                  06496000 D
                                                          %                          06497000 D
                                                          %                          06498000 D
                                   OP433:                 % RX TYPE JUMP, LINK MEMORY 06499000 D
                                                          % (P) + 2 INTO (Y); (Y) + 1 INTO P  06500000 D
                                                          % B HOLDS THE INSTRUCTION   06501000 D
13E7  570C 00C0 0030 C060          RXMFIELD - 1 = CPCR     % (Y) IN B                 06502000 D
13E8  48C9 0C41 1010 00F0          B L = BR2, A3                                      06503000 D
13E9  000C 00C0 0030 C030          COMP 8 = SAR                                       06504000 D
13EA  4809 A001 200C 00F0          A1 L = A2              % PUT PAR INTO A2            06505000 D
13EB  002C 00C0 0C30 00A0          COMP 16 = SAR; 2 = LIT                             06506000 D
13EC  4809 C0C0 A00C 00F0          A2 R = A2              % (P) INTO A2               06507000 D
13ED  48C9 C140 C030 C0FC          A2 + LIT = MIR         % MIR CONTAINS (P) + 2       06508000 D
13EE  61EC 00C0 0C00 0060          EMULOUT - 1 = CPCR     % (P) + 2 INTO (Y)           06509000 D
13EF  4809 E0C0 9000 C0FC          A3 R = A3              % (Y) INTO A3                06510000 D
13F0  000C C0C0 0030 C010          8 = SAR                                            06511000 D
13F1  4809 E0C0 0B30 00F0          A3 + 1 = B             % (Y) + 1 INTO B             06512000 D
13F2  48C9 A000 C030 00F0          A1 R = A1                                          06513000 D
13F3  0000 0000 0000 0020          16 = SAR                                           06514000 D
13F4  4809 A001 4000 00F0          A1 L = A1              % A1 READY TO ACCEPT NEW PAR 06515000 D
13F5  4809 AC5C 400C 00F0          A1 OR B = A1           % CREATE NEW PAR             06516000 D
13F6  66EC 00C0 0000 C040          OPCODE - 1 = MPCR                                  06517000 D
                                                          %                          06518000 D
                                                          %                          06519000 D
                                   OPCODE44:              % JUMP ZERO                 06520000 D
13F7  5F90 CC00 0030 0060          XECODE - 1 = CPCR      % RETURNS "F" CODE IN A      06521000 D
13F8  4809 C640 0040 C0FC          A2 + AMPCR = AMPCR                                 06522000 D
13F9  1D70 0000 0030 00C0          OP44F - 1 = AMPCR                                  06523000 D
13FA  4809 0003 0030 C0FC          STEP                                               06524000 D
13FB  4824 0000 0000 00F0          EXEC                                               06525000 D
                                                          %                          06526000 D
                                                          %                          06527000 D
13FC  1D8C 00C0 0C30 0040          OP44F:  OP440 - 1 = MPCR                           06528000 D
13FD  1D90 00C0 0000 C040          OP441 - 1 = MPCR                                   06529000 D
13FE  1DA0 0000 0C00 0040          OP442 - 1 = MPCR                                   06530000 D
13FF  1DB0 0000 0030 0040          OP443 - 1 = MPCR                                   06531000 D
                                                          %                          06532000 D
                                                          %                          06533000 D
                                   OP440:                 % RR TYPE JUMP ZERO         06534000 D
                                                          % IF (R(A)) = 0, (R(M)) IN P  06535000 D
1400  22BC C0C0 0000 C060          CONTENTSRA - 1 = CPCR  % RETURNS (R(A)) IN B       06536000 D
1401  4809 CC52 0030 C0F0          B EQL 0                                            06537000 D
```

```
1402  6819 0000 0000 00F0          IF TRUE THEN SKIP                                  06538000 D
1403  66E0 0000 0000 0040          OPCODE - 1 = MPCR                                  06539000 D
1404  2380 00C0 0000 0060          CONTENTSRM - 1 = CPCR  % RETURNS (R(M)) IN B       06540000 D
1405  4809 A000 C030 C0FC          A1 R = A1              % CLEAR OLD PAR             06541000 D
1406  00CC 00C0 0C30 002C          COMP 16 = SAR                                      06542000 D
1407  4809 A0C1 4C3C 00F0          A1 L = A1                                          06543000 D
1408  4809 AC5C 400C 00FC          A1 OR B = A1           % CREATE NEW PAR            06544000 D
1409  66E0 00C0 003C 0040          OPCODE - 1 = MPCR                                  06545000 D
                                   %                                                  06546000 D
                                   %                                                  06547000 D
                     OP441:        % RI TYPE 1 LOCAL JUMP EQUAL                       06548000 D
                                   % IF (CC) = OR 0, (P) + D INTO P                   06549000 D
140A  2250 0000 0070 0060          CHECKCC - 1 = CPCR     % RETURNS CC BITS IN B      06550000 D
140B  4809 0C52 0C30 00F0          B EQL 0                                            06551000 D
140C  6819 C000 0030 00F0          IF TRUE THEN SKIP                                  06552000 D
140D  66E0 C0C0 C00C C040          OPCODE - 1 = MPCR                                  06553000 D
140E  5590 0000 0000 0C40          RI1 - 1 = MPCR         % (P) + D = P               06554000 D
                                   %                                                  06555000 D
                                   %                                                  06556000 D
                     OP442:        % RX TYPE JUMP ZERO                                06557000 D
                                   % IF (R(A)) 0, Y INTO P                            06558000 D
140F  2280 C000 0C00 0060          CONTENTSRA - 1 = CPCR  % GET (R(A)) INB            06559000 D
1410  4809 0C52 003C 00F0          B EQL 0                                            06560000 D
1411  6819 0000 003C 00F0          IF TRUE THEN SKIP                                  06561000 D
1412  1030 0000 0000 C040          BUMP - 1 = MPCR        % ADVANCE THE PAR BY 1 AND CALL OPCODE  06562000 D
1413  4839 E156 1C30 0CF0          A3 AND LIT = A3        % "M" FIELD IN A3           06563000 D
1414  00F0 0003 003C 00EC          15 = LIT                                           06564000 D
1415  48C9 EC12 0000 C0FC          A3 EQL 0                                           06565000 D
1416  6C19 00C0 0E30 00FC          IF TRUE THEN 0 = B; SKIP                           06566000 D
1417  2380 00C0 0000 0060          CONTENTSRM - 1 = CPCP  % RETURN (R(M)) IN B        06567000 D
1418  48C9 CC40 100C 00F0          B = A3                                             06568000 D
1419  633C 0000 003C 0060          IFETCH - 1 = CPCR      % Y INTO B                  06569000 D
141A  4809 EC40 0B30 00FC          A3 + B = B             % PAR VALUE IN LS WORD OF E  06570000 D
141B  4809 A003 C000 C0FC          A1 R = A1              % CLEAR OLD PAR             06571000 D
141C  0000 C0C0 0C30 002C          COMP 16 = SAR                                      06572000 D
141D  4809 A001 403C 00F0          A1 L = A1                                          06573000 D
141E  48C9 AC5C 4030 00F0          A1 OR B = A1           % CREATE NEW PAR            06574000 D
141F  66E0 0000 003C C04C          OPCODE - 1 = MPCR                                  06575000 D
                                   %                                                  06576000 D
                                   %                                                  06577000 D
                     OP443:        % RX TYPE JUMP ZERO                                06578000 D
                                   % IF (R(A)) = 0, (Y) INTO P                        06579000 D
1420  2280 0000 0000 C060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B             06580000 D
1421  4809 0C52 C030 C0F0          B EQL 0                                            06581000 D
1422  6819 0000 0C00 C0F0          IF TRUE THEN SKIP                                  06582000 D
1423  103C 0000 0000 C040          BUMP - 1 = MPCR        % ADVANCE THE PAR BY 1 AND CALL OPCODE  06583000 D
1424  4809 EC03 0B3C 00F0          A3 = B                                             06584000 D
1425  5700 0000 0000 C06C          RXMFIELD - 1 = CPCR    % ANALYZE THEN "M" FIELD    06585000 D
1426  4809 CC41 0C1C 00FC          B L = BR2              % PUT Y INTO BR2             06586000 D
1427  00C0 0003 0C30 CC30          COMP 8 = SAR                                       06587000 D
1428  60EC 00C0 0C30 C060          EMULIN - 1 = CPCR      % (Y) INTO B                06588000 D
1429  4809 A003 C000 C0FC          A1 R = A1              % CLEAR OLD PAR             06589000 D
142A  0000 0000 000C C020          COMP 16 = SAR                                      06590000 D
142B  4809 AC01 4000 C0FC          A1 L = A1                                          06591000 D
142C  4809 AC5C 4C30 00F0          A1 OR B = A1           % CREATE NEW PAR            06592000 D
142D  66E0 0000 0030 C04C          OPCODE - 1 = MPCR                                  06593000 D
                                   %                                                  06594000 D
                                   %                                                  06595000 D
                     OPCODE45:     % JUMP NOT ZERO                                    06596000 D
142E  5F90 C000 0C30 0060          XFCODE - 1 = CPCR      % RETURN "F" FIELD IN LSD OF A2  06597000 D
```

```
142F  4809 C640 0090 00F0              A2 + AMPCR = AMPCR
1430  10C0 C000 000C 00C0              OP45F - 1 = AMPCR
1431  4809 0000 0000 C0F0              STEP
1432  4824 00C0 000C C0F0              EXEC
                                                           *
                                                           *
1433  10DC C000 C000 C040   OP45F:     OP45C - 1 = MPCR
1434  10E0 00C0 0C0C 004C              OP451 - 1 = MPCR
1435  10FC 0000 0030 C040              OP452 - 1 = MPCR
1436  1EPC C003 0C30 C040              OP453 - 1 = MPCR
                                                           *
                                                           *
                            OP450:                         * RR TYPE JUMP NOT ZERO
                                                           * IF (R(A)) NE 0, (R(M)) INTO P
1437  22BC 0000 0C30 006C              CONTENTSRA - 1 = CPCR   * (R(A)) INTO B
1438  4809 0C52 C000 00F0              B EOL 0
1439  580E 00C0 C030 00F0              IF TRUE THEN STEP ELSE SKIP
143A  56E0 0CC0 00DC 0040              OPCODE - 1 = MPCR
143B  238C 0CC0 0C30 0060              CONTENTSRM - 1 = CPCR   * RETURN (R(M)) IN B
143C  4809 A000 CC3C 00F0              A1 R = A1
143D  0000 000D 0C3C 0020              COMP 16 = SAR
143E  4809 A0C1 400C 00F0              A1 L = A1               * A1 PAR FIELD CLEARED
143F  4809 AC5C 4000 C0FC              A1 OR B = A1            * CREATE NEW PAR
1440  56EC C0C0 C000 C040              OPCODE - 1 = MPCR
                                                           *
                                                           *
                            OP451:                         * RI TYPE JUMP NOT EQUAL
                                                           * IF (CC) NE, (P) + D INTO P
1441  226C 00C0 CCDC C060              CHECKCC - 1 = CPCR      * PUT CC BITS IN B
1442  4809 0C52 0C30 0CF0              B EOL 0
1443  6806 0000 CC3C C0F0              IF TRUE THEN STEP ELSE SKIP
1444  66EC 00C0 CC30 0G4C              OPCODE - 1 = MPCR
1445  5590 0003 000C C040              RI1 - 1 = MPCR          * (P) + D = P
                                                           *
                                                           *
                            OP452:                         * RK TYPE JUMP NOT ZERO
                                                           * IF (R(A)) NE 0, Y INTO P
1446  2280 00C3 0000 C060              CONTENTSRA - 1 = CPCR   * (R(A)) INTO B
1447  4809 0C52 C03C C0F0              B EOL 0
1448  680E 00C0 CC30 C0F0              IF TRUE THEN STEP ELSE SKIP
1449  103C C0C0 0000 0040              BUMP - 1 = MPCR         * ADVANCE THE PAR BY 1 AND CALL OPCODE
144A  4809 E155 1030 C0F0              A3 AND LIT = A3         * "M" FIELD IN A3
144B  00FC C0C0 0030 00E0              15 = LIT
144C  4809 EC12 0C0C 00FC              A3 EOL 0
144D  6C19 0003 0BDC 60F0              IF TRUE THEN 0 = B; SKIP
144E  238C 0000 000C 0060              CONTENTSRM - 1 = CPCR   * (R(M)) IN B
144F  4809 6C40 1030 00FC              P = A3
1450  533C C0C0 000C C060              IFETCH - 1 = CPCR       * PUT Y INTO B
1451  4809 EC40 0B00 00F0              A3 + B = B
1452  4609 A0C3 C030 00F0              A1 R = A1
1453  0000 C0C0 0030 002C              COMP 16 = SAR
1454  4809 A001 400C C0FC              A1 L = A1
1455  48C9 AC5C 4000 00F0              A1 OR D = A1            * CREATE NEW PAR
1456  56EC 00C0 0C30 0040              OPCODE - 1 = MPCR
                                                           *
                                                           *
                            OP453:                         * RX TYPE JUMP NOT ZERO
                                                           * IF (R(A)) NE 0, (Y) INTO P
1457  4809 CC40 8B0C 00F0              B R = B                 * SHIFT "A" TO LS BITS
1458  8CFC C003 CCC0 C080              4 = SAR; 15 = LIT
```

```
1459  4809 2C56 0800 00E0          LIT AND B = B          % ISOLATE "A"                    06658C00 D
145A  51C0 0FC0 0000 C060          REGSTACK - 1 = CPCR     % R(A) INTO MAR2                06659C00 D
145B  2F30 0FC0 0C0C C060          EINPUT - 1 = CPCR      % (R(A)) INTO B                  06660C00 D
145C  4809 0C52 000C 00E0          B EQL 0                                                 06661C00 D
145D  500B 0C00 0030 00FC          IF TRUE THEN STEP ELSE SKIP                             06662C0C D
145E  1D3C 0000 0030 004C          BUMP - 1 = MPCR        % ADVANCE THE PAR BY 1 AND CALL OPCODE 06663C00 D
145F  57D0 C003 0C30 0060          RXMFIELD - 1 = CPCR    % ANALYZE "M" FIELD              06664C00 D
1460  4809 0C41 0010 00FC          D L = BR2              % ADDRESS OF Y INTO BR2           06665C0C D
1461  0000 C0C0 000C C03C          COMP B = SAR                                            06666C00 D
1462  60E0 C000 0C3C C060          EMULIN - 1 = CPCR      % (Y) IN B                       06667C00 D
1463  48C9 A0C0 C000 00E0          A1 R = A1                                               06668C0C D
1464  000C 00C0 00C0 0020          COMP 16 = SAR                                           06669C00 D
1465  4809 A0C1 403C C0E0          A1 L = A1                                               0667CC00 D
1466  4809 AC5C 4C0C 00E0          A1 OR B = A1           % CREATE NEW PAR                 06671C00 D
1467  66E0 C0C3 0C00 0040          OPCODE - 1 = MPCR                                       06672C00 D
                                                        %                                06673C00 D
                                                        %                                06674C0C D
                                OPCODE46:               % JUMP POSITIVE                   0667SC00 D
1468  5F9C 000C 000C 00E0          XFCODE - 1 = CPCR      % GET "F" CODE IN A2              06676C00 D
1469  4809 C640 0030 00FC          A2 + AMPCR = AMPCR                                      06677C00 D
146A  1E10 000C 0C0C 00C0          OP46F - 1 = AMPCR                                       0667BC0C D
146B  4809 0000 0030 00FC          STEP                                                    06679C00 D
146C  4824 C00C 0C3C C0F0          EXEC                                                    0668CP00 D
                                                        %                                06681C00 D
                                                        %                                06682C00 D
146D  1E20 0000 0C3C 0040       OP46F:  OP460 - 1 = MPCR                                   06683C3C D
146E  1E3C 00C3 0C0C 004C          OP461 - 1 = MPCR                                        06684C00 D
146F  1E40 00C0 003C 0040          OP462 - 1 = MPCR                                        06685C00 D
1470  1E50 0000 0000 C04C          OP463 - 1 = MPCR                                        06686C00 D
                                                        %                                06687C0C D
                                                        %                                06688C0C D
                                OP460:                  % RR TYPE JUMP POSITIVE          06689C00 D
                                                        % IF (R(A)) >= 0, (R(M)) INTO P   0669CC00 D
1471  228C 000C 0C3C C060          CONTENTSRA - 1 = CPCR % (R(A)) INTO B                   06691C00 D
1472  4809 CC41 0B0C 00E0          B L = B                % SHIFT (R(A)) TO MS WORD         06692C0C D
1473  300C C0C3 0000 0020          COMP 16 = SAR                                           06693C0C D
1474  4809 0C4D 0C3C C0F0          P                      % PUT B IN THE ADDER             06694C00 D
1475  480E 00C0 0C0C 00F0          IF MST THEN STEP ELSE SKIP  % TEST (R(A)) >= 0          06695C00 D
1476  66E0 0CC0 000C 0040          OPCODE - 1 = MPCR                                       06696C00 D
1477  2380 0CC0 000C 0060          CONTENTSRM - 1= CPCR   % (R(M)) INTO B                  06697CC0 D
1478  4809 A0C0 CCCC C0FC          A1 R = A1              % CLEAR THE PAR FIELD             06698C00 D
1479  3000 00C3 0030 0020          COMP 16 = SAR                                           06699C00 D
147A  4809 A0C1 403C 00F0          A1 L = A1                                               067CCC00 D
147D  4009 AC5C 4C0C 00F0          A1 OR B = A1           % SET UP NEW PAR                 06703C00 D
147C  66E0 C000 C000 0040          OPCODE - 1 = MPCR                                       06702C00 D
                                                        %                                06703C0C D
                                                        %                                06704C00 D
                                OP461:                  % R1 TYPE 1 LOCAL JUMP >=         0670SC00 D
147D  226C 0000 0000 C06C          CHECKCC - 1 = CPCR     % RETURN CC IN LS BITS OF B      06706C30 D
147E  4809 2C52 0C3C C0F0          LIT EQL B              % TEST FOR <                     06707C0C D
147F  0030 0CC3 0030 00EC          3 = LIT                                                 06708C00 D
1480  68CB 0000 0000 CCFC          IF TRUE THEN STEP ELSE SKIP                             06709000 D
1481  66EC 0C00 0030 0040          OPCODE - 1 = MPCR                                       C671CC00 D
1482  559C 0003 0C0C 0040          R11 - 1 = MPCR         % (P) + D = P                    C6711C00 D
                                                        %                                06712CC0 D
                                                        %                                06713C0C D
                                OP462:                  % RK TYPE JUMP POSITIVE           06714CC0 D
                                                        % IF (R(A)) >= 0, Y INTO P        06715C0C D
1483  228C CCC3 0000 006C          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                  06716C00 D
1484  4809 CC41 0B3C 00F0          B L = B                % SHIFT (R(A)) INTO MS WOPE OF B  06717C0C D
```

```
1485  0000 0000 0020 0020          COMP 16 = SAR                                  06718000 D
1486  4809 0C40 0000 00F0          B                      % PUT B IN THE ADDER    06719000 D
1487  480E 0000 0070 00F0          IF MST THEN STEP ELSE SKIP                     06720000 D
1488  1D30 0000 0000 0040          BUMP - 1 = MPCR        % ADVANCE THE PAR BY 1 AND CALL OPCODE  06721000 D
1489  4809 E156 1C70 00F0          A3 AND LIT = A3        % "M" FIELD IN A3       06722000 D
148A  00E0 0000 0000 00E0          15 = LIT                                       06723000 D
148B  4809 E012 0030 C0F0          A3 EOL 0                                       06724000 D
148C  6C19 0000 0B0C 00FC          IF TRUE THEN 0 = B; SKIP                       06725000 D
148D  2380 0000 C000 0060          CONTENTSRM - 1 = CPCR  % (R(M)) INTO 9         06726000 D
148E  4809 0C40 1000 00F0          B = A3                                         06727000 D
148F  633C 0000 0C00 C06C          IFETCH - 1 = CPCR      % Y INTO B              06728000 D
1490  4809 EC40 CB00 00F0          A3 + B = B             % PUT PAR IN LS WORD OF B  06729000 D
1491  4809 ACC0 C00C 00F0          A1 R = A1                                      06730000 D
1492  0000 0000 0C0C C020          COMP 16 = SAR                                  06731000 D
1493  480C 4001 4C00 00FC          A1 L = A1              % CLEAR PAR FROM PSW     06732000 D
1494  4809 AC5C 4C00 C0FC          A1 OR B = A1                                   06733000 D
1495  66EC 00C0 0000 C040          OPCODE - 1 = MPCR                              06734000 D
                                                        %                        06735000 D
                                                        %                        06736000 D
                                   OP463:               % RX JUMP POSITIVE       06737000 D
                                                        % IF (R(A)) >= 0, (Y) INTO P  06736000 D
1496  4809 0C40 8B00 C0F0          B R = B                % "A" INTO LS BITS        06739000 D
1497  80F0 0000 0000 C080          4 = SAR; 15 = LIT                              06740000 D
1498  4809 2C56 0B00 C0F0          LIT AND B = B          % ISOLATE "A" INTO B     06741000 D
1499  51C0 C0C3 0C00 C06C          REGSTACK - 1 = CPCR    % R(A) INTO MAR2        06742000 D
149A  2F3C 0000 0000 C060          EINPUT - 1 = CPCR      % (R(A)) INTO E         06743000 D
149B  4809 0C41 0B00 C0FC          B L = B                % SHIFT (R(A)) INTO MS WORD OF B  06744000 D
149C  0000 C0C3 0C30 6020          COMP 16 = SAR                                  06745000 D
149D  4809 0C40 0000 00F0          B                      % PUT B IN THE ADDER    06746000 D
149E  480B C000 0C30 C0F0          IF MST THEN STEP ELSE SKIP                     06747000 D
149F  1D30 0000 0000 C04C          BUMP - 1 = MPCR        % ADVANCE THE PAR BY 1 AND CALL OPCODE  06748000 D
14A0  4809 E003 0B00 C0F0          A3 = B                                         06749000 D
14A1  570C 0C00 0C0C 0060          PXMFIELD - 1 = CPCR                            06750000 D
14A2  4809 0C41 0010 00F0          B L = BR2              % PUT Y INTO BR2        06751000 D
14A3  0C00 C0C0 003C C030          COMP 8 = SAR                                   06752000 D
14A4  60EC 00C3 0C00 0060          EMULIN - 1 = CPCR      % PUT (Y) INTO B        06753000 D
14A5  4809 A00D C000 C0FC          A1 R = A1              % CLEAR OLD PAR          06754000 D
14A6  0000 00C0 0C00 C020          COMP 16 = SAR                                  06755000 D
14A7  480B A001 4C00 C0FC          A1 L = A1                                      06756000 D
14A8  4809 AC5C 4C00 C0FC          A1 OR B = A1           % CREATE NEW PAR         06757000 D
14A9  66EC 00C0 0C00 0040          OPCODE - 1 = MPCR                              06758000 D
                                                        %                        06759000 D
                                                        %                        06760000 D
                                   OPCODE47:            % JUMP NEGATIVE INSTRUCTION  06761000 D
14AA  5F9C 0000 0000 C06C          XFCODE - 1 = CPCR      % RETURNS "F" FIELD IN LSB OF A2  06762000 D
14AB  4809 C640 0C40 00F0          A2 + AMPCR = AMPCR                             06763000 D
14AC  1E6C C0C3 0C30 C0C0          OP47F - 1 = AMPCR                              06764000 D
14AD  4809 0CC3 0C0C C0F0          STEP                                           06765000 D
14AE  4824 0C00 00C0 00FC          EXEC                   % JUMP TO DESIRED FORMAT TYPE  06766000 D
                                                        %                        06767000 D
                                                        %                        06768000 D
14AF  1E70 0000 0030 C04C          OP47F: OP470 - 1 = MPCR                        06769000 D
14B0  1E8C 0000 0C00 0040          OP471 - 1 = MPCR                               06770000 D
14B1  1E9C C000 0C00 0040          OP472 - 1 = MPCR                               06771000 D
14B2  1EA0 C000 0030 0040          OP473 - 1 = MPCR                               06772000 D
                                                        %                        06773000 D
                                                        %                        06774000 D
                                   OP470:               % FR FORMAT JUMP REGISTER NEGATIVE  06775000 D
                                                        % IF (R(A)) < 0, (R(M)) INTO P  06776000 D
14B3  228C 00C3 0C70 0060          CONTENTSRA - 1 = CPCR  % (R(A)) INTO B         06777000 D
```

```
14B4  4809 CC41 0C30 C0F0          B L = B                  % SHIFT (R(A)) INTO UHW OF B      06778C0C D
14B5  0000 00C3 0C30 C02C          COMP 16 = SAR                                             06779C0C D
14B6  4809 0C40 0000 00F0          B                        % PUT B IN THE ADDER, TEST (R(A)) < 0  0678 0C00 D
14B7  4819 C003 C03C 00FC          IF MST THEN SKIP                                          0678 1C00 D
14B8  66EC 0CC3 CC3C C040          OPCODE - 1 = MPCR                                         06782C00 D
14B9  2380 0000 0030 C06C          CONTENTSRM - 1 = CPCR    % (R(M)) INTO B                  06783C30 D
14BA  4809 ACC3 CC3C 00F0          A1 R = A1                % CLEAR LOWER 16 BITS OF A1       0678 4C 0 D
14BB  000C 00C3 0C3C 0020          COMP 16 = SAR                                             06785C00 D
14BC  4809 ACC1 4C30 00F0          A1 L = A1                                                 06786000 D
14BD  4809 AC5C 4030 C0F0          A1 OR B = A1             % MODIFY PAR TO JUMP              06787C00 C
14BE  56EC 0000 0030 CC40          OPCODE - 1 = MPCR                                         0678 8C3C D
                                                            %                                06789C00 D
                                                            %                                06790C00 D
                              OP471:                        % R1 TYPE I LOCAL JUMP LESS INSTRUCTIOF  06791C00 C
                                                            % IF (CC) = 11, (P) + 0 = F       06792CC0 D
14BF  2809 0000 0C3C C0F0          IF LC1                                                    06793C0C D
14C0  226C C0C3 0C3C 0C6C          CHECKCC - 1 = CPCR       % PUT THE CC BITS INTO B          06794C00 D
14C1  4809 2C52 000C C0F0          LIT FOL B                % CHECK CC FOR 11                 06795C00 D
14C2  003C C0C3 000C 00E0          3 = LIT                                                   06796C00 D
14C3  6CCB C0C3 C000 C0FF          IF FALSE THEN STEP ELSE SKIP                              06797C00 D
14C4  66EC 00C0 0C30 0040          OPCODE - 1 = MPCR                                         06798CC0 D
14C5  5590 0000 0000 0C4C          R11 - 1 = MPCR           % (P) + 0 = P                    06799C00 D
                                                            %                                068C0000 D
                                                            %                                068C1C0C D
                              OP472:                        % RK TYPE JUMP NEGATIVE           068C2000 D
                                                            % IF (R(A)) < 0, (Y) = P          068C3C0C D
14C6  22BC 00C3 0030 0060          CONTENTSRA - 1 = CPCR    % (R(A)) INTO B                   068C4C00 D
14C7  4809 0C41 CB30 C0F0          B L = B                  % SHIFT (R(A)) INTO UHW OF B       068C5C00 D
14C8  00C0 00C0 003C 0020          COMP 16 = SAR                                             068C6C0C D
14C9  4809 0C40 00FC CCFC          B                        % CHECK IF (R(A)) < 0             068C7C00 D
14CA  4819 00C3 0030 C0F0          IF MST THEN SKIP                                          068C8C00 D
14CB  103C C0C3 0C3C C04C          BUMP - 1 = MPCR          % ADVANCE THE FAR BY 1 AND CALL OPCODE  068C9C0C D
14CC  2380 60C3 000C C060          CONTENTSRM - 1 = CPCR    % (R(M)) INTO E                  06811C0C D
14CD  4809 0C40 1000 00F0          B = A3                                                    06811C00 D
14CE  633C 0CC3 0030 006C          IFETCH - 1 = CPCR        % PUT Y INTO B                   06812CC0 D
14CF  4009 EC40 0B0C 00FC          A3 + B = B               % PUT PAR VALUE IN LS WORD OF B   06813C00 C
14D0  480C A0C3 C000 00FC          A1 R = A1                % CLEAR OLD PAR                   06814C00 D
14D1  0C0C 00C0 003C 002C          COMP 16 = SAR                                             06815C00 D
14D2  4809 A001 4000 00FC          A1 L = A1                                                 06816C30 D
14D3  480C AC5C 4C30 C0FC          A1 OP B = A1             % CONSTRUCT NEW PAR               06817C00 D
14D4  56EC 00C0 003C 0040          OPCODE - 1 = MPCR                                         0681 8C00 C
                                                            %                                06819C00 D
                                                            %                                0692CC00 D
                              OP473:                        % RX TYPE JUMP INSTRUCTION        06821C00 D
                                                            % IF (R(A)) < 0, (Y) = P          06822C00 D
14D5  4809 0C40 8B30 C0FC          E R = B                  % SHIFT "A" INTO LS BITS OF B      06823C00 D
14D6  30F0 C0C0 003C 0080          4 = SAR, 15 = LIT                                         06824C0C D
14D7  4809 2C56 0B3C C0F0          LIT AND B = B            % ISOLATE "A" INTO B              06825C00 D
14D8  5100 00C0 C00C C06C          REGSTACK - 1 = CFCR      % R(A) INTO MAR2                  06826C00 D
14D9  2F30 0C00 0000 006C          EINPUT - 1 = CPCR        % (R(A)) INTO E                   06827C00 D
14DA  4809 0C41 0B0C 00F0          B L = B                  % (R(A)) SHIFTED INTO UHW OF B    06828C00 D
14DB  0000 00C0 0C00 C02C          COMP 16 = SAR                                             06829C00 C
14DC  4809 0C40 003C C0FC          B                        % CHECK IF (R(A)) < 0             0683 0C3C D
14DD  4819 00C0 0C30 00F0          IF MST THEN SKIP                                          06831C00 D
14DE  103C 0CC0 CC0C C04C          BUMP - 1 = MPCR          % ADVANCE THE FAR BY 1 AND CALL OPCODE  06832C00 D
14DF  4809 E00C 0B3C C0FC          A3 = B                   % RESTORE INSTRUCTION IN B        06833C0C D
14E0  570C 00C3 003C C06C          RXMFIELD - 1 = CFCR      % ANALYZE "M" FIELD               06834C00 D
14E1  4809 0C41 CC1C C0F0          E L = BR2                % PUT Y INTO BR2                   06835CC0 D
14E2  000C 00C3 0000 0C30          COMP 8 = SAR                                              06836C0C D
14E3  56E0 C0C0 0C0C CC60          EMULIN - 1 = CPCR        % PUT (Y) INTO B                  06837C00 D
```

```
14E4  4809 A003 C030 00F0          A1 R = A1                    % CLEAR THE OLD PAR                  06838C00 D
14E5  0000 0000 0C30 0020          COMP 16 = SAR                                                     06839C00 D
14E6  4809 A001 4000 00F0          A1 L = A1                                                         06840C00 D
14E7  4809 AC5C 4000 00F0          A1 OR B = A1                 % PUT (Y) INTO THE PAR                06841C00 D
14E8  66EC 0000 0000 0C40          OPCODE = 1 = MPCR                                                 06842C00 D
                                                                %                                    06843C00 D
                                                                %                                    06844C00 D
                                   OPCODE50:                    % NOT IMPLEMENTED                     06845000 D
14E9  4E5C 0000 0030 0040          NOTIMP = 1 = MPCR                                                 06846C00 D
                                                                %                                    06847C00 D
                                   OPCODE51:                    % NOT IMPLEMENTED                     06848C00 D
14EA  4E5C 0000 0030 0040          NOTIMP = 1 = MPCR                                                 06849C00 D
                                                                %                                    06850C00 D
                                   OPCODE52:                    % NOT IMPLEMENTED                     06851000 D
14EB  4E50 0003 0030 004C          NOTIMP = 1 = MPCR                                                 06852C00 D
                                                                %                                    06853000 D
                                   OPCODE53:                    % NOTIMPLEMENTED                      06854C00 D
14EC  4E50 0000 0000 0040          NOTIMP = 1 = MPCR                                                 06855C00 D
                                                                %                                    06856C0C D
                                                                %                                    06857C00 D
                                   OPCODE54:                    % LOAD ADDRESS REGISTER(S)            06858C00 D
14ED  5F90 0003 0070 C06C          XFCODE = 1 = CPCR                                                 06859C00 D
14EE  4809 C640 0C40 0CF0          A2 + AMPCR = AMPCR                                                06860C00 D
14EF  1EBC 0003 C030 00C0          OP54F = 1 = AMPCR                                                 06861C00 D
14F0  4809 0000 003C 00FC          STEP                                                              06862C00 D
14F1  4824 0000 0000 00F0          EXEC                                                              06863C00 D
                                                                %                                    06864C00 D
14F2  1ECC 00C0 0C00 0040          OP54F:   OP540 - 1 = MPCR                                         06865C00 D
14F3  1ED0 C003 0000 0040          OP541 - 1 = MPCR                                                  06866C00 D
14F4  300C 0C00 007C 0040          FAULT - 1 = MPCR                                                  06867C00 D
14F5  1EE0 0000 0000 0040          OP543 - 1 = MPCR                                                  06868C00 D
                                                                %                                    06869C0C D
                                   OP540:                       % LARR: LOAD ADDRESS REGS: TYPE RR   06870C00 D
14F6  4809 E155 0B00 00F0          A3 AND LIT = B                                                    06871C00 D
14F7  00FC 0000 0000 C0E0          15 = LIT                                                          06872C00 D
14F8  51C0 0000 0000 006C          REGSTACK - 1 = CPCR                                               06873C00 D
14F9  2F30 C003 0000 0060          EINPUT - 1 = CPCR            % B = (R(M))                         06874C00 D
14FA  4809 0C40 0030 C0F0          B = MIR                                                           06875C0C D
                                                                %                                    06876C0C D
14FB  4809 E000 8B90 00F0          LAR:     A3 R = D                                                 06877C00 D
14FC  90F0 0000 0C00 0080          4 = SAR; 15 = LIT                                                 06878C0C D
14FD  4809 2C56 CB30 CCF0          LIT AND B = B                % B = R(A)                           06879C00 D
14FE  51C0 0003 0000 0060          REGSTACK - 1 = CPCR                                               06880C0C D
14FF  2F30 0003 000C 0C6C          EINPUT - 1 = CPCR            % B = (R(A))                         06881C00 D
1500  48C9 2C56 2000 C0F0          LIT AND B = A2               % A2 = (R(A))7-0                      06882C00 D
1501  0FFC 00C0 0C0C 00E0          255 = LIT                    % 8 BIT MASK                          06883C00 D
1502  4809 C643 001C C0F0          A2 + AMPCR = MAR2            % MAR2 = AR                           06884C00 D
1503  3000 G000 000C 00C0          PAGEREG = AMPCR              % ADD EASE ADDR + A2                  06885C00 D
1504  4809 00C0 0C30 C0F0          STEP                         % SEPARATE AMPCR ASSIGNMENTS          06886C0C D
1505  2F50 00C0 003C C06C          EOUTPUT - 1 = CPCR                                                06887C0C D
1506  66EC 00C3 0C00 0040          OPCODE = 1 = MPCR                                                 06888C00 D
                                                                %                                    06889C0C D
                                   OP541:                       % LARI: LOAD ADDRESS REGISTER (INDIRECT) 0689CC0C D
                                                                % (Y*) -> AR R                       06891C00 D
1507  4809 E156 0B30 C0F0          A3 AND LIT = B                                                    06892C00 D
1508  00FC 00C0 C000 C0EC          15 = LIT                                                          06893C00 D
1509  51C0 00C0 C000 0060          REGSTACK - 1 = CPCR                                               06894C00 D
150A  2F30 0000 0000 0060          EINPUT - 1 = CPCR            % B = (R(M)) = Y*                    06895C0C D
150B  4809 0C41 9010 C0F0          D L = BR2                    % Y* INTO BR2                        06896C00 D
150C  0000 0000 0030 C030          COMP 8 = SAR                                                      06897C00 D
```

```
1500  50EC C0C0 0000 0060          EMULIN - 1 = CPCR      % B = (Y*)                    06898C00 D
150E  4809 CC40 CC3C 00FC          B = MIR                                              06899C00 D
150F  4FA0 C000 0C00 005D          LAR - 1 = MPCR                                       06900C00 D
                                                          %                            0690 1C00 D
                                   OP543:                 % LARM; LOAD ADDRESS REG. MULTIPLE 06902C00 D
                                                          %(Y,...Y+U) -> AR P...AR R+U 06903C00 D
                                                          % MULTIPLE LOADS OBSERVE PAGE 0690 4C00 D
                                                          % ADDRESSES                   0690 5C00 D
1510  57C0 0003 C0C0 C06D          RXMFIELD - 1 = CPCR    % B = Y                       0690 6C00 D
1511  4809 0C40 0C9C C0F0          B = MIR                                              0690 7C00 D
1512  4809 E0C0 8B0C C0FD          A3 R = B                                             0690 8C00 D
1513  80FC 0000 0C00 00B0          4 = SAR; 15 = LIT                                    0690 9C00 D
1514  4809 2C56 0B3C 00F0          LIT AND B = B          % E = :A: FIELD               0691 0C00 D
1515  51C0 0000 0000 C06C          REGSTACK - 1 = CPCR                                  0691 1C00 D
1516  2F30 C003 0C0C 0D60          EINPUT - 1 = CPCR      % B = (R(A))                  0691 2C00 D
1517  0FF0 0000 0000 0D9C          255 = LIT ; B = SAR    % A2 = PAGE REG NUM.           0691 3C00 D
1518  4809 2C56 203C C0FD          LIT AND B = A2                                       0691 4C00 D
1519  4809 C640 1070 C0FD          A2 + AMPCR = A3        % A3 = AR R                    0691 5C00 D
151A  3C0C 0000 0000 CCCC          PAGEREG = AMPCR                                      0691 6C00 D
151B  49C9 0C40 8B00 00F0          B R = B; SET LC1       % B HAS COUNT: LC1 SET FOR R +1 MOVE 0691 7C00 D
151C  4809 0C52 0000 C0F0          C EOL B                % COUNT OF ZERO CAUSES ALL REG TO BE 0691 8C00 D
1510  9FFC C0C0 002C 0DA0          255 = LIT; COMP 16 = SAR                             0691 9C00 D
151E  5C19 2001 0B3C C0FD          IF TRUE THEN LIT + 1 L = B: SKIP                     0692 0C00 D
151F  4809 0C47 C03C C0FD          B + 1 L = B                                          0692 1C00 D
1520  4809 EC5C 103C C0F0          A3 OR B = A3,BM1       % COUNT IN UNW OF A3          0692 2C00 D
1521  4809 0C41 C02C 0DF0          B L = BR1              % ORIGIN ADDR IN BR1           0692 3C00 D
1522  0000 0000 0C00 0030          COMP B = SAR                                         0692 4C00 D
1523  4A2C 0CC0 0030 C06D          MOVE - 1 = CPCR                                      0692 5C00 D
1524  66EC 00C0 0C0C 0040          OPCODE - 1 = MPCR                                    0692 6C00 D
                                                          %                            0692 7C00 D
                                                          %                            0692 8C00 D
                                   OPCODE55:              % STORE ADDRESS REGISTER(S)    0692 9C00 D
1525  5F9C 0000 CC3C C060          XFCODE - 1 = CPCR                                    0693 0C00 D
1526  4809 C640 0040 00FC          A2 + AMPCR = AMPCR                                   0693 1C00 D
1527  1EFC 0003 0000 CCC0          OP55F - 1 = AMPCR                                    0693 2C00 D
1528  4809 00C0 0030 00F0          STEP                                                 0693 3C00 D
1529  4824 C0C0 0C3C 00FD          EXEC                                                 0693 4C00 D
                                                          %                            0693 5C00 D
152A  1F0C 0CC0 CC3C C04C          OP55F: OP550 - 1 = MPCR                              0693 6C00 D
152B  1F10 00C0 0C30 C040          OP551 - 1 = MPCR                                     0693 7C00 D
152C  3000 00C0 003C C040          FAULT - 1 = MPCR                                     0693 8000 D
152D  1F20 0000 0000 0040          OP553 - 1 = MPCR                                     0693 9C00 D
                                                          %                            0694 0C0C 0
                                   OP55D:                 % SARR; STORE ADDRESS REGISTER (REGISTER)0694 1C00 D
                                                          % TYPE RR: (AR) A -> RM       0694 2C00 D
152E  4809 0C40 8B0C 0DFC          B R = B                                              0694 3C00 D
152F  80FC 0003 C00C C0B0          4 = SAR; 15 = LIT                                    0694 4C00 D
1530  4809 2C56 0B3C 00F0          LIT AND B = B                                        0694 5C00 D
1531  51C0 00C0 C0D0 C06D          REGSTACK - 1 = CPCR    % MAR2 = RA                   0694 6C00 D
1532  2F30 0CC0 CC30 0060          EINPUT - 1 = CPCR      % B = (R(A))                  0694 7C00 D
1533  48C9 2C56 203C 00FC          LIT AND B = A2         % A2 = REG NUM.               0694 8C00 D
1534  0FFC 00C0 0C0C 0CE0          255 = LIT                                            0694 9C00 D
1535  4809 C640 CC1C C0F0          A2 + AMPCR = MAR2                                    0695 0C00 D
1536  3000 0000 0C00 CDC0          PAGEREG = AMPCR        % PAGEREG EASE + REG #        0695 1C00 D
1537  4809 00C3 C0C0 C0FC          STEP                   % SEPARATE AMPCR ASSIGNMENTS  0695 2C0C 0
1538  2F30 CCC0 CC30 0060          EINPUT - 1 = CPCR      % B = (AR)                    0695 3C0C D
1539  4809 0C40 0C3C 0CF0          B = MIR                                              0695 4C00 D
153A  4809 E156 0B00 00F0          A3 AND LIT = B         % B = :M: FIELD               0695 5C00 D
153B  0CFC CCC0 0030 C0E0          15 = LIT                \                            0695 6C00 D
153C  51C0 00C0 0C30 C06C          REGSTACK - 1 = CPCR    % MAR2 = RM                   0695 7C00 D
```

```
153D  2F5C 0000 0000 0060          EOUTPUT - 1 = CPCR    % (AR) -> RM                  06958C00 D
153E  66E0 00C0 0000 0040          OPCODE - 1 = MPCR                                   06959C00 D
                                                         %                             06960C00 D
                                   OP551:                % SAR1; STORE ADDRESS REGISTER 06961C00 D
                                                         % TYPE R1; (AR) A -> Y*       06962C00 D
153F  4809 0C43 8B30 C0F0          B R = B                                             06963C00 D
1540  80F0 00C3 0030 0080          4 = SAR; 15 = LIT                                   06964C00 D
1541  4809 2C56 CB00 C0F0          LIT AND B = B         % D = :A: FIELD               06965C00 D
1542  51C0 00C0 00D0 0060          REGSTACK - 1 = CPCR   % MAR2 = R#                   06966C00 D
1543  2F30 00D3 0030 0060          EINPUT - 1 = CPCR     % B = (R(A))                  06967C00 D
1544  4809 2C56 2000 C0F0          LIT AND B = A2        % A2 = REG #                   06968C00 D
1545  0FFC 0000 003C 00E0          255 = LIT                                           06969C00 D
1546  4809 C643 001C C0F0          A2 + AMPCR = MAR2                                   06970C00 D
1547  3000 00D0 0C30 00CC          PAGEREG = AMPCR       % PAGEREG BASE + REG #        06971C00 D
1548  4809 00C0 0C30 00F0          STEP                  % SEPARATE AMPCR ASSIGNMENTS  06972C00 D
1549  2F3C 00D0 0030 0060          EINPUT - 1 = CPCR     % B = (AR)                    06973C00 D
154A  4809 0C40 0030 00F0          B = MIR                                             06974C00 D
154B  4809 E156 0B30 00F0          A3 AND LIT = B        % D = :M: FIELD               06975C00 D
154C  00FC 00C0 0000 C0E0          15 = LIT                                            06976C00 D
154D  51CC 0000 CC0C 0060          REGSTACK - 1 = CPCR                                 06977C00 D
154E  2F30 0CC3 0C0C 0060          EINPUT - 1 = CPCR     % D = (R(M)) = Y*             06978C00 D
154F  4809 0C41 0C1C 00F0          P L = BR2             % Y* INTO BR2                 06979C00 D
1550  000C 00C0 000C 0030          COMP B = SAR                                        06980C00 D
1551  51EC 00C0 0C0C 0060          EMULOUT - 1 = CPCR    % (AR) -> Y*                  06981C00 D
1552  66E0 C0C0 0000 0040          OPCODE - 1 = MPCR                                   06982C00 D
                                                         %                             06983C00 D
                                   OP553:                % STORE ADDRESS REGISTER MULTIPLE 06984C00 D
                                                         % SARM; TYPE RX (AR R,...AR R+U) -> 06985C00 D
                                                         % Y,...Y+U                    06986C00 D
                                                         % MULTIPLE STORES DO OBSERVE PAGE 06987C00 D
                                                         % ADDRESSES                   06988000 D
1553  5700 CCC0 0030 0060          RXMFIELD - 1 = CPCR                                 06989C00 D
1554  4809 CC40 0030 C0FC          B = MIR               % MIR = Y                     06990000 D
1555  4809 E0C0 8B00 00F0          A3 R = B                                            06991C00 D
1556  80F0 CCC0 00D0 C08C          4 = SAR; 15 = LIT                                   06992C00 D
1557  4809 2C56 0B0C 00F0          LIT AND B = B         % D = :A: FIELD               06993C00 D
1558  51C0 00C3 0000 0060          REGSTACK - 1 = CPCR                                 06994C00 D
1559  2F30 00C0 0030 0060          EINPUT - 1 = CPCR     % D = (R(A))                  06995C00 D
155A  4809 2C56 2000 C0FC          LIT AND B = A2        % A2 HAS PAGE REG #           06996C00 D
155B  0FF0 0000 0000 0090          255 = LIT; 8 = SAR                                  06997C00 D
155C  4809 C640 1000 C0FC          A2 + AMPCR = A3       % P = AR R                    06998C00 D
155D  3000 00C3 00C0 00CC          PAGEREG = AMPCR                                     06999000 D
155E  4809 0C40 8B30 00F0          E R = B               % SHIFT COUNT INTO LS BITS OF P 07000C00 D
155F  4809 0C52 0C00 00F0          0 EOL B               % A COUNT OF ZERO MEANS ALL REG STORED 07001C00 D
1560  0FF0 0C0D 0C3C C0AC          255 = LIT; COMP 16 = SAR                            07002C00 D
1561  6C19 20C1 0B0C 00F0          IF TRUE THEN LIT + 1 L = B; SKIP                    07003C00 D
1562  4809 CC47 0B0D 00F0          B + 1 L= B            % COUNT IN B                  07004C00 D
1563  4809 E001 0020 00F0          A3 L = BR1            % ORIGIN ADDR IN BR1          07005000 D
1564  0000 0CC0 00CC C030          COMP B = SAR                                        07006000 D
1565  4809 0C40 1D3C 00F0          B = A3;BM1            % COUNT IN UHW OF A3          07007C00 D
1566  4809 EC5C 1000 00F0          A3 OR B = A3          % DESTINATION ADDR IN LHW OF A3 07008C00 D
1567  4A2C 00C0 C030 0060          MOVE - 1 = CPCR                                     07009C00 D
1568  66E0 C0C0 0C0C C040          OPCODE - 1 = MPCR                                   07010C00 D
                                                         %                             07011C00 D
                                                         %                             07012C00 D
                                   OPCODF56:             % NOT IMPLEMENTED             07013C0C D
1569  4E5C 00C0 0000 0040          NOTIMP - 1 = MPCR                                   07014C00 D
                                                         %                             07015C0C D
                                   OPCODE57:             % NOT IMPLEMENTED             07016C00 D
156A  4E50 0CC0 C030 004C          NOTIMP - 1 = MPCR                                   07017C00 D
```

```
                                                    *                                         07018C00 D
156B  5F90 0000 0000 0060   OPCODE60:  XFCODE - 1 = CPCR                                         07019C00 D
156C  4809 C640 0030 00F0              A2 + AMPCR = AMPCR                                         07020C00 D
156D  1F30 0000 0000 00C0              OP60F - 1 = AMPCR                                          07021000 D
156E  4809 0000 0030 C0F0              STEP                                                       07022C00 D
156F  4824 0000 0000 C0F0              EXEC                                                       07023C00 D
                                                    *                                         07024C00 D
1570  1F40 0000 0000 0040   OP60F:     OP600 - 1 = MPCR                                           07025C00 D
1571  1F50 0000 0000 C040              OP601 - 1 = MPCR                                           07026C00 D
1572  1F60 0000 0000 C040              OP602 - 1 = MPCR                                           07027C00 D
1573  1F70 0000 0000 C040              OP603 - 1 = MPCR                                           07028C00 D
                                                    *                                         07029000 D
                                                    *                                         07030C00 D
                            OP600:                  * RL TYPE LOGIC RIGHT SINGLE SHIFT        07031C00 D
                                                    * SHIFT (R(A)) RIGHT M (0-3), SET CC,     07032C00 D
1574  22B0 00C0 0000 0060              CONTENTSRA - 1 = CPCR * AND ZEROFILL                       07033C00 D
1575  4809 E156 1000 00F0              A3 AND LIT = A3      * "M" INTO A3                          07034000 D
1576  00FC 00C0 0030 00E0              15 = LIT                                                   07035C00 D
1577  4809 E000 0000 80F0              A3 = SAR                                                   07036C0C D
1578  4809 CC40 8030 00F0              B R = MIR, B                                               07037C0C D
1579  2000 0000 0000 0060              SETCCA - 1 = CPCR    * SET THE CONDITION BITS              07038C00 D
157A  2F5C 0000 0000 0060              EOUTPUT - 1 = CPCR                                         07039C00 D
157B  66EC 0000 000C 0040              OPCODE - 1 = MPCR                                          07040C00 D
                                                    *                                         07041C00 D
                                                    *                                         07042C00 D
                            OP601:                  * RL TYPE ALG RIGHT SINGLE SHIFT          07043C00 D
                                                    * (R(A)) RIGHT M (0-3), SIGN FILL, SET CC07044C0C D
157C  2809 0000 0000 C0F0              IF LC1                                                     07045C00 D
157D  22BC C0C0 0000 0060              CONTENTSRA - 1 = CPCR, * (R(A)) INTO B                     07046C00 D
157E  4809 0C00 2000 00F0              C = A2                                                     07047C0C D
157F  4809 0C41 0E30 C0F0              B L = B                                                    07048C00 D
1580  000C 0000 0000 C020              COMP 16 = SAR                                              07049C0C D
1581  4809 0C40 0000 C0FC              B                                                          07050C00 D
1582  4C09 C018 2000 00F0              IF MST THEN B111 = A2                                      07051C00 D
1583  4809 E156 1000 00F0              A3 AND LIT = A3                                            07052C00 D
1584  00FC C0C0 000C 00A0              15 = LIT; COMP 16 = SAR                                    07053C00 D
1585  4809 CCC1 2000 C0F0              A2 L = A2            * A2 = 000C/C0C0 OR 1111/C00C         07054C00 D
1586  4809 CC40 8B00 00FC              B R = B                                                    07055C0C D
1587  4809 CC5C 0B0C C0F0              A2 OR B = B          * B = 1111/(R(A)) OR 0000/(R(A))      07056C00 D
1588  4809 E000 0000 80FC              A3 = SAR                                                   07057C00 D
1589  4809 0C40 8B00 00F0              B R = B              * PERFORM ALG RIGHT SINGLE SHIFT      07058C0C D
158A  4809 CC41 0B0C C0F0              B L = B              * CLEAR UHW OF B                      07059C0C D
158B  0000 0000 0000 0020              COMP 16 = SAR                                              07060C0C D
158C  4809 CC40 8B3C 00F0              B R = D, MIR                                               07061C0C D
158D  2F5C 0000 0000 0060              EOUTPUT - 1 = CPCR                                         07062C00 D
158E  200C C0C0 0000 C060              SETCCA - 1 = CPCR    * SET THE CONDITION BITS              07063C00 D
158F  66EC 0000 0C00 C040              OPCODE - 1 = MPCR                                          07064C00 D
                                                    *                                         07065C00 D
                                                    *                                         07066000 D
                            OP602:                  * RL TYPE LOGICAL RIGHT DOUBLE SHIFT      07067C00 D
                                                    * (R(A),R(A+1)) M (0-3) PLACES, ZERO FILLC7068C00 D
                                                    * AND SET CC                              07069C0C D
1590  4809 2C56 200C C0F0              LIT AND B = A2       * ISOLATE "M"                         07070C00 D
1591  90F0 00C0 0030 008C              15 = LIT; 4 = SAR                                          07071C0C D
1592  4809 0C40 8B30 C0F0              B R = B                                                    07072C00 D
1593  4809 2C56 0B0C 00FC              LIT AND B = B        * OBTAIN "A" FIELD                    07073CC0 D
1594  51CC 0000 0000 C060              REGSTACK - 1 = CPCR                                        07074C00 D
1595  4809 0F41 0020 00F0              BMAR L = BR1                                               07075C00 D
1596  0000 C0C0 0C30 C030              COMP 8 = SAR                                               07076C00 D
1597  2F30 0000 000C 0060              EINPUT - 1 = CPCR    * (R(A)) INTO B                       07077C00 D
```

```
1598  4809 0C41 1000 00F0          B L = A3                                                  07078C00 D
1599  001C 0000 0000 00A0          16 = SAR / 1 = LIT                                        07079000 D
159A  4809 2F5C 001C 00FC          LIT OR BMAR = MAR2      % R(A+1)                          0708CC00 D
159B  2F30 0C03 0C0C C060          EINPUT - 1 = CPCR       % (R(A+1)) INTO B                 07081C00 D
159C  4809 EC5C 100C 00F0          A3 OR B = A3            % A3 = (R(A))/(R(A+1))             07082C00 D
159D  4809 CCCC C03C 80FC          A2 = SAR                                                  07083C00 D
159E  49C9 E000 9B3C 00FC          A3 R = A3,BI SET LC1    % PERFORM DOUBLE SHIFT            07084C00 D
159F  20DC 0000 C00C 006C          SETCCA - 1 = CPCR       % SET THE CONDITION BITS           07085C00 D
15A0  4809 E001 0B3C 00F0          A3 L = B                                                  07086C00 D
15A1  0000 0000 C00C C02C          COMP 16 = SAR                                             07087C0C D
15A2  4809 0C40 903C C0F0          B R = MIR               % NEW (R(A+1))                    07088C00 D
15A3  2F50 0003 0000 0060          EOUTPUT - 1 = CPCR      % WRITE NEW R(A+1)                07089C0C D
15A4  4809 0000 CCCC 20F0          ASR                     % REFERENCE BR1                   07090C0C D
15A5  4809 0F43 801C 00FC          BMAR R = MAR2           % R(A)                            07091C00 D
15A6  0000 0CCC 0000 C010          B = SAR                                                   07092C00 D
15A7  4809 E003 8C3C 00F0          A3 R = MIR              % (R(A))                          07093C00 D
15A8  000C C003 000C C02C          16 = SAR                                                  07094C00 D
15A9  2F5C 0CC0 000C 0C60          EOUTPUT - 1 = CPCR                                        07095C00 D
15AA  56E0 6003 0C3C 0040          OPCODE - 1 = MPCR                                         07096C00 D
                                                           %                                07097C00 D
                                                           %                                07098C00 D
                              0F6C3:                       % RL TYPE ALG RIGHT DOUBLE SHIFT  07099C0C D
                                                           % RIGHT SHIFT (R(A),R(A+1)), M (0-3) 0710CC00 D
                                                           % SIGN FILL AND SET CC            07101C00 D
15AB  4809 2C56 203C C0FC          LIT AND B = A2          % ISOLATE "M" FIELD               07102C0C D
15AC  00F0 C30C 0C3C C0E0          15 = LIT                                                  07103C00 D
15AD  28C9 6C40 8B3C C0FC          B R = BI IF LC1                                           07104C00 D
15AE  80FC 00CC 0C3C C080          4 = SARI 15 = LIT                                         07105C00 D
15AF  4809 2C56 0B0C 00F0          LIT AND B = 6                                             07106C00 D
15B0  51C0 0CC3 0000 0C60          REGSTACK - 1 = CPCR      % R(A) INTO MAR2                 07107CC0 D
15B1  4809 0F41 0020 00F0          BMAR L = BR1            % TEMP STORAGE                    07108C00 D
15B2  0000 6003 0C3C 0030          COMP 8 = SAR                                              07109C00 D
15B3  2F3C 0003 0C3C 0060          EINPUT - 1 = CPCR       % (R(A)) INTO B                   07110C0C D
15B4  4809 0C41 0B0C C0F0          B L = B                                                   07111C0C D
15B5  000C 0003 0C3C 0C2C          COMP 16 = SAR                                             07112C00 D
15B6  4809 0C40 0000 C0F0          B                       % PUT B IN THE ADDER              07113000 D
15B7  4BC9 CCCC 0000 C0F0          IF MST THEN SET LC1     % NEG SIGN FLAG                   07114C0C D
15B8  0C10 0000 0C0C C0A0          COMP 16 = SAR / 1 = LIT                                   07115C00 D
15B9  4809 EC5C 1C0C 00FC          A3 OR B = A3            % A3 = (R(A))/INSTRUCTION         07116C00 D
15BA  4809 2F5C 001C C0F0          LIT OR BMAR = MAR2      % R(A+1)                          07117C0C D
15BB  2F3C 00C3 0C0C C0A0          EINPUT - 1 = CPCR       % (R(A+1)) INTO B                 07118C00 D
15BC  4809 ECC3 9C0C 00F0          A3 R = A3               % CLEAR LHW OF A3                 0711AC00 D
15BD  0000 0CC3 0C0C 0020          16 = SAR                                                  07120C00 D
15BE  4809 E001 1030 00FC          A3 L = A3                                                 07121C00 D
15BF  4809 EC5C 1030 C0F0          A3 OR B = A3            % A3 = (R(A))/(R(A+1))            07122C00 D
15C0  4809 CCCC 0C30 80F0          A2 = SAR                                                  07123C00 D
15C1  4809 E003 9030 00FC          A3 R = A3               % PERFORM SHIFT                   07124C00 D
15C2  4809 0000 0B0C 00F0          0 = B                                                     07125C00 D
15C3  2C09 0013 0B30 C0F0          IF LC1 THEN B111 = B                                      07126FC0 D
15C4  4809 0C41 CB0C 00FC          B L = B                 % B = 1111/0000 OR B = 0000/C000  07127C0C D
15C5  49C9 EC5C 1E30 00F0          A3 OR B = A3,BI SET LC1                                   07128CC0 D
15C6  20DC 0CC3 0000 C060          SETCCA - 1 = CPCR       % SET THE CONDITION BITS          07129C0C D
15C7  4809 EC01 2C30 C0FC          A3 L = A2                                                 07130C00 D
15C8  0000 0CC3 0C3C C02C          COMP 16 = SAR                                             07131C0C D
15C9  4809 CCC0 8C30 CCFC          A2 R = MIR              % (R(A+1)) SHIFTED INTO MIR       07132C00 D
15CA  2F50 0C03 0C3C C060          EOUTPUT - 1 = CPCR                                        07133C00 D
15CB  4809 C003 0C3C 2CF0          ASR                     % REFERNCE BR1                    07134C00 D
15CC  4809 0F43 801C 00FC          BMAR R = MAR2           % R(A)                            07135C00 D
15CD  000C 0C03 C00C 001C          B = SAR                                                   07136C0C D
15CE  4809 E0C3 8C3C 00F0          A3 R = MIR                                                07137C00 D
```

```
15CF  9000 0000 0000 0020          16 = SAR                                                    07138000 0
15D0  2F5C 0000 0000 C060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A))                   07139000 0
15D1  66EC 0000 0000 C040          OPCODE - 1 = MPCR                                           07140000 0
                                   %                                                           07141000 0
                                   %                                                           07142000 0
15D2  5F9C 0000 0000 D060  OPCODE61:  XFCODE - 1 = CPCR   % "F" INTO A2                        07143000 0
15D3  4809 C640 0090 00F0          A2 + AMPCR = AMPCR                                          07144000 0
15D4  1F80 0000 0000 C0C0          OP61F - 1 = AMPCR                                           07145000 0
15D5  4809 CCC0 C090 C0FC          STEP                                                        07146000 0
15D6  4824 0003 0090 C0F0          EXEC                                                        07147000 0
                                   %                                                           07148000 0
15D7  1F9D C003 000C D040  OP61F:  OP610 - 1 = MPCR                                            07149000 0
15D8  1FAC 0060 000C C04C          OP611 - 1 = MPCR                                            07150000 0
15D9  1FBC 0CC0 0000 C040          OP612 - 1 = MPCR                                            07151000 0
15DA  1FCC 0F00 0000 0040          OP613 - 1 = MPCR                                            07152000 0
                                   %                                                           07153000 0
                                   %                                                           07154000 0
                           OP610:                  % RL TYPE ALG LEFT SINGLE SHIFT           07155000 0
                                                   % (R(A)) M (0-3) PLACES, ZERO FILL        07156000 0
                                                   % SET CC AND SET OVERFLOW BIT             07157000 0
15DB  4809 2C56 203C C0F0          LIT AND B = A2         % "M" FIELD INTO A2                   07158000 0
15DC  8CF0 CCC0 C090 C080          15 = LIT; 4 = SAR                                          07159000 0
15DD  4809 CC40 8000 0CF0          B R = B                                                    07160000 0
15DE  4809 2C56 0B00 C0F0          LIT AND B = B          % ISOLATE "A"                       07161000 0
15DF  51CC 0CC0 0C9C C060          REGSTACK - 1 = CPCR                                        07162000 0
15E0  2F3C 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO E                     07163000 0
15E1  4809 CCC2 2030 C0F0          NOT A2 = A2                                                07164000 0
15E2  4809 C0C0 0000 80F0          A2 + 1 = SAR                                               07165000 0
15E3  4809 0C41 0096 00F0          B L = B                % PERFORM LEFT SHIFT                 07166000 0
15E4  4809 CC40 A00C C0F0          B R = A2                                                   07167000 0
15E5  0008 0000 0000 0020          16 = SAR                                                   07168000 0
15E6  4809 C002 0030 C0F0          NOT A2                                                     07169000 0
15E7  63C9 0000 000C C0F0          IF NOT ABT THEN SET LC1                                    07170000 0
15E8  5020 00C0 C00C 0060          OVBIT - 1 = CPCR       % SET THE OVERFLOW BIT              07171000 0
15E9  4809 0C41 0B3C 00F0          B L = B                                                    07172000 0
15EA  0000 0C00 G00C C020          COMP 16 = SAR                                              07173000 0
15EB  4809 0C40 8B3C 00F0          B R = MIR, B           % LEFT SHIFTED (R(A))               07174000 0
15EC  2F5C 0C00 0C0C 0060          EOUTPUT - 1 = CPCR     % WRITE NEW (R(A))                  07175000 0
15ED  200C 0000 009C C060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS            07176000 0
15EE  66E0 0000 0C00 C040          OPCODE - 1 = MPCR                                          07177000 0
                                   %                                                          07178000 0
                                   %                                                          07179000 0
                           OP611:                  % TYPE RL CIRCULAR LEFT SHIFT             07180000 0
                                                   % (R(A)) SHIFTED M (0-3) PLACES, SET CC   07181000 0
15EF  4809 2C56 203C 00F0          LIT AND B = A2         % "M" INTO A2                        07182000 0
15F0  8CF0 CCC0 9C0C 008C          15 = LIT; 4 = SAR                                          07183000 0
15F1  4809 0C40 8B3C 00F0          B R = B                                                    07184000 0
15F2  4809 2C56 0B00 C0FC          LIT AND B = B          % ISOLATE "A" FIELD                 07185000 0
15F3  51C0 00C0 003C 0060          REGSTACK - 1 = CPCP                                        07186000 0
15F4  2F3C 0CC0 003D C060          EINPUT - 1 = CPCR      % (R(A)) INTO B                     07187000 0
15F5  4809 0C40 1030 00FC          B = A3                                                     07188000 0
15F6  4809 E0C1 100C CCF0          A3 L = A3                                                  07189000 0
15F7  0100 0000 0000 00AC          COMP 16 = SAR; 16 = LIT                                    07190000 0
15F8  4809 EC5C 103C 00F0          A3 OR B = A3           % A3 = (R(A))/(R(A))                07191000 0
15F9  4809 C000 CB00 C0F0          A2 = B                 % "M" INTO B                        07192000 0
15FA  4809 2C5E 2000 C0F0          LIT - B = A2           % 16 COMPLEMENT OF SHIFT AMOUNT      07193000 0
15FB  4809 C000 000C 80F0          A2 = SAR                                                   07194000 0
15FC  4809 E0C1 9000 C0F0          A3 C = A3              % PERFORM SHIFT                      07195000 0
15FD  4809 E001 103D 00FC          A3 L = A3                                                  07196000 0
15FE  0600 0CC0 CC3C CC20          COMP 16 = SAR                                              07197000 0
```

```
15FF  4809 E000 8830 00F0          A3 R = MIR, E                                  07198000 D
1600  2F50 0003 0040 C060          EOUTPUT - 1 = CPCR     % WRITE OUT NEW R(A))   07199000 D
1601  2C00 0003 0000 C060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS 07200000 D
1602  56EC 0000 0000 0040          OPCODE - 1 = MPCR                              07201000 D
                                                          %                      07202000 D
                                                          %                      07203000 D
                                   OP612:                 % RL TYPE ALG LEFT DOUBLE SHIFT 07204000 D
                                                          % (R(A),R(A+1)), H (0-3)       07205000 D
                                                          % ZERO FILL, SET CC, AND SET OVERFLOW 07206000 D
1603  4809 0040 8800 00F0          B R = B                                        07207000 D
1604  80F0 0000 0000 0080          4 = SAR, 15 = LIT                              07208000 D
1605  4809 E156 2000 00F0          A3 AND LIT = A2        % "H" INTO A2           07209000 D
1606  4809 2056 0B00 00F0          LIT AND B = B          % ISOLATE "A" FIELD     07210000 D
1607  51C0 0000 0000 0060          REGSTACK - 1 = CPCR                            07211000 D
1608  4809 0F41 0020 00F0          BMAR L = BR1           % TEMP STORAGE OF R(A)  07212000 D
1609  0000 0000 0000 0030          COMP 8 = SAR                                   07213000 D
160A  2F30 0000 0000 C060          EINPUT - 1 = CPCR      % (R(A)) INTO B         07214000 D
160B  4809 0041 1000 00F0          B L = A3                                       07215000 D
160C  0010 0000 0000 00A0          COMP 16 = SAR, 1 = LIT                         07216000 D
160D  4809 2F5C 001C 00F0          LIT OR BMAR = MAR2     % R(A+1)                07217000 D
160E  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B       07218000 D
160F  48C9 EC5C 1000 00F0          A3 OR B = A3           % A3 =(R(A))/(R(A+1))   07219000 D
1610  4809 C002 2000 00F0          NOT A2 = A2                                    07220000 D
1611  4809 C0C0 0030 80F0          A2 + 1 = SAR                                   07221000 D
1612  4809 E000 8800 00F0          A3 R = B                                       07222000 D
1613  4809 0042 0000 00F0          NOT B                                          07223000 D
1614  63C9 0003 0020 00F0          IF NOT ABT THEN SET LC1                        07224000 D
1615  5020 0000 0000 C060          OVBIT - 1 = CPCR       % SET THE OVERFLOW BIT  07225000 D
1616  4809 C0C0 0000 80F0          A2 + 1 = SAR                                   07226000 D
1617  49C9 E001 1B00 00F0          A3 L = A3, BJ SET LC1                          07227000 D
1618  2000 0000 0070 C060          SETCCA - 1 = CPCR      % SET THE CONDITION BITS 07228000 D
1619  4809 E001 0B30 00F0          A3 L = B                                       07229000 D
161A  0000 0000 0000 0020          COMP 16 = SAR                                  07230000 D
161B  48C9 0040 8090 00F0          B R = MIR                                      07231000 D
161C  2F50 0003 0070 C060          EOUTPUT - 1 = CPCR     % OUTPUT (R(A+1))       07232000 D
161D  4809 C0C0 0000 20F0          ASR                    % REFERENCE BR1         07233000 D
161E  48C9 0F41 8010 C0F0          BMAR R = MAR2          % R(A)                  07234000 D
161F  0000 00C0 CC90 C010          8 = SAR                                        07235000 D
1620  48C9 E0C0 8C90 00F0          A3 R = MIR                                     07236000 D
1621  0000 0003 0000 C020          16 = SAR                                       07237000 D
1622  2F50 0000 0000 0060          EOUTPUT - 1 = CPCR     % OUTPUT (R(A))         07238000 D
1623  56EC 0000 C000 0040          OPCODE - 1 = MPCR                              07239000 D
                                                          %                      07240000 D
                                                          %                      07241000 D
                                   OP613:                 % RL TYPE CIRCULAR LEFT DOUBLE SHIFT 07242000 D
                                                          % (R(A),R(A+1)), H (0-3) AND SET CC  07243000 D
1624  4809 2056 2000 00F0          LIT AND B = A2         % "H" INTO A2           07244000 D
1625  80F0 0000 0C00 0080          15 = LIT, 4 = SAR                              07245000 D
1626  4809 0040 8800 00F0          B R = B                                        07246000 D
1627  4809 2056 0B0C 00F0          LIT AND B = B          % "A" INTO B            07247000 D
1628  51C0 00C0 0000 0060          REGSTACK - 1 = CPCR                            07248000 D
1629  4809 0F41 0020 00F0          BMAR L = BR1           % TEMP STORAGE          07249000 D
162A  0000 0000 0000 C030          COMP 8 = SAR                                   07250000 D
162B  2F30 0003 0000 0060          EINPUT - 1 = CPCR      % (R(A)) INTO B         07251000 D
162C  4809 0041 1C30 00F0          B L = A3                                       07252000 D
162D  0010 0000 0000 00A0          COMP 16 = SAR, 1 = LIT                         07253000 D
162E  4809 2F5C 001C C0FC          LIT OR BMAR = MAR2     % R(A+1)                07254000 D
162F  2F30 0000 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B       07255000 D
1630  4809 EC5C 1000 00F0          A3 OR B = A3           % A3 = (R(A))/(R(A+1))  07256000 D
1631  4809 C002 2000 00F0          NOT A2 = A2                                    07257000 D
```

```
1632  4809 C0C0 0C30 80F0          A2 + 1 = SAR                                  07258C00 D
1633  49C9 E001 9B00 C0F0          A3 C = A3,D; SET LC1  % CIRCULAR SHIFT         07259000 D
1634  2C00 0000 0F0C 006C          SETCCA - 1 = CPCR     % SET THE CONDITION BITS 07260C00 D
1635  4809 E001 0B30 00F0          A3 L = B                                      07261C00 D
1636  000C 0000 0000 C020          COMP 16 = SAR                                 07262C00 D
1637  4809 0C40 8C30 C0F0          B R = MIR                                     07263C0C D
1638  2F50 00C0 0C00 0060          EOUTPUT - 1 = CPCR    % OUTPUT (R(A+1))       07264C00 D
1639  4809 0000 0CD0 20F0          ASR                   % REF DR1               07265C0C D
163A  4809 CF40 801C 00F0          RMAR R = MAR2                                 07266C00 D
163B  000C C0C3 0C30 C010          B = SAR                                       07267C0C D
163C  4809 E0C0 803C 00FC          A3 R = MIR            % (R(A)) INTO MIR        07268C00 D
163D  000C 00C3 C030 0020          16 = SAR                                      07269C00 D
163E  2F5C C0C0 0C00 CC60          EOUTPUT - 1 = CPCR    % OUTPUT SHIFTED (R(A))  07270CCC D
163F  56E0 00C0 0C30 0040          OPCODE - 1 = MPCR                             07271C00 D
                                                         %                       07272C00 D
                                                         %                       07273C00 D
1640  5F9C C0C0 0030 0060    OPCODE62: XFCODE - 1 = CPCR % "F" INTO A2            07274CC0 C
1641  4809 C640 0C30 00F0          A2 + AMPCR = AMPCR                            07275C00 D
1642  1FDC 00C0 C030 00C0          OP62F - 1 = AMPCR                             07276C00 D
1643  4809 0000 0C3C C0FC          STEP                                          07277C00 D
1644  4824 0000 0C30 00F0          EXEC                                          07278C3C D
                                                         %                       07279C00 D
1645  1FEC 0000 003C C040    OP62F:  OP620 - 1 = MPCR                            07280C00 D
1646  1FFC C003 0C30 0040          OP621 - 1 = MPCR                              07281C00 C
1647  2000 00C0 0000 0040          OP622 - 1 = MPCR                              07282C0C D
1648  2010 00C0 0C0C 0040          OP623 - 1 = MPCR                              07283C00 D
                                                         %                       07284000 D
                                                         %                       07285C00 C
                                   OP620:                % TYPE RL SUBTRACT, (R(A))-M=R(A)  07286C00 D
                                                         % SET CC, CARRY AND OV BITS  07287C00 D
1649  22BC CCC0 0C00 C060          CONTENTSKA - 1 = CPCR % (R(A)) INTO B          07288C00 D
164A  4809 CC41 2C3C 00FC          B L = A2              % (R(A)) INTO MS WORD OF A2  07289C0C D
164B  00F0 00C3 0006 C0A0          COMP 16 = SAR; 15 = LIT                       0729CC00 D
164C  4639 E156 0B30 C6F0          A3 AND LIT = B        % "M" INTO B             07291C0C D
164D  4809 0C41 0B2C C0F0          B L = B               % "M" INTO MS WORD OF B   07292C0C D
164E  4649 CC5E 0C30 00FC          A2 - B = MIR; SET LC2                         07293C00 D
164F  7BC9 C000 C03C C0F0          IF AOV THEN SET LC1                           07294C00 L
1650  1E70 C0C0 0C30 006C          CARRY - 1 = CPCR                              07295C00 D
1651  4F1C 00C3 CC30 C060          CHECKOV - 1 = CPCR                            07296C00 D
1652  4809 00C0 0D3C 00FC          RMI                                           07297C0C D
1653  4809 0C40 8B3C C0F0          B R = MIR,B           % NEW (R(A)) INTO LS WORD OF MIR,B  07298C00 D
1654  00C0 C0C0 0C2C C02C          16 = SAR                                      07299C00 D
1655  20DC C0C0 C03C 0060          SETCCA - 1 = CPCR     % SET THE CONDITION BITS 073CCC0C D
1656  2F5C C0C0 0C3C 0060          EOUTPUT - 1 = CPCR    % (R(A))-M INTO R(A)     07301C00 D
1657  56EC 0000 00FC CC4C          OPCODE - 1 = MPCR                             07302C00 D
                                                         %                       07303C0C D
                                                         %                       07304C00 D
                                   OP621:                % TYPE RL LITERAL SUBTRACT DOUBLE  07305C00 D
                                                         %(R(A),R(A+1)) - M = R(A),R(A+1)  07306C00 D
                                                         % SET CC, CARRY AND OV BITS  07307C0C D
1658  4809 2C56 0C30 00FC          LIT AND B = MIR       % "M" INTO MIR            07308C00 D
1659  80F0 0CC0 0C30 0080          15 = LIT; 4 = SAR                             07309C00 D
165A  4809 0C40 88CC C0FC          B R = B               % SHIFT "A" TO LS 4 BITS  07310C0C D
165B  4809 2C55 0B3C C0F0          LIT AND B = D         % ISOLATE "A" FIELD       07311C00 D
165C  51CC C0C3 0C3C C060          REGSTACK - 1 = CPCR                           07312C0C D
165D  4809 0F41 0020 C0F0          RMAR L = DR1          % TEMP STORAGE           07313CCC D
165E  000C C0C0 0C00 0030          COMP 8 = SAR                                  07314C00 D
165F  2F30 C0C0 CC0C 0060          FINPUT - 1 = CPCR     % (R(A)) INTO B          07315C0C D
1660  4809 0C41 2C00 C0F0          B L = A2              % (R(A)) INTO MS WORD OF A2  07316C0C D
1661  001C 00C3 0C00 00A0          COMP 16 = SAR + 1 = LIT                       07317C0C D
```

```
1662  4809 2F5C 0C1C 00F0          LIT OR BMAR = MAR2     % R(A+1)                    07318C00 D
1663  2F3C 00C0 0000 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B           07319C00 D
1664  4809 CC5C 2D9C 00F0          A2 OR B = A2, BMI      % A2 = (R(A))/(R(A+1))       07320D00 D
1665  4849 CC5E 009C 00F0          A2 - 9 = MIR; SET LC2                              07321C00 D
1666  78C9 0000 0000 00F0          IF AOV THEN SET LC1                                07322C00 D
1667  1E7C 0003 009C 0060          CARRY - 1 = CPCR                                   07323C00 D
1668  4F1C 0003 0C9C 0060          CHECKOV - 1 = CPCR                                 07324C00 D
1669  49C9 0003 0D0C C0F0          BMI; SET LC1                                       07325C00 D
166A  2000 0003 009C 006C          SETCCA - 1 = CPCR                                  07326C0C D
166B  48F9 0003 0D9C C0F0          BMI                                                07327D00 D
166C  4809 0C41 2D90 C0F0          B L = A2                                           07328C00 D
166D  0000 0FC9 0C0C C020          COMP 16 = SAR                                      07329C00 D
166E  4809 CC40 8B9C 00F0          B R = B                                            07330C00 D
166F  4809 C0C0 8C9C 00FC          A2 R = MIR             % NEW (R(A+1))              07331D0C D
1670  2F50 00C3 0C0C 0060          EOUTPUT - 1 = CPCR                                 07332C00 D
1671  4809 0003 CC9C 20F0          ASR                    % PER BR1                   07333D09 D
1672  48C9 0F43 801C 00F0          BMAR R = MAR2                                      07334CC0 D
1673  00F0 0000 0C00 C010          B = SAR                                            07335C30 D
1674  4809 CC40 009C 00F0          B = MIR                % NEW (R(A))                 07336D00 D
1675  2F50 0000 0000 C060          EOUTPUT - 1 = CPCR                                 07337D0C D
1676  66EC 0000 0D90 C040          OPCODE - 1 = MPCR                                  07338C00 D
                                                        %                           07339C0C D
                                                        %                           07340C00 D
                                   OP622:               % TYPE RL LITERAL ADD, (R(A))+M -> R(A)  07341D00 D
                                                        % SET CC, CARRY AND OV BITS  07342D00 D
1677  22BC 0C03 F000 0060          CONTENTSRA - 1 = CPCR % (R(A)) INTO B              07343C00 D
1678  4809 CC41 2D90 00F0          B L = A2                                           07344C00 D
1679  0CF0 C000 0090 00A0          COMP 16 = SAR; 15 = LIT                            07345C00 D
167A  4809 E155 0B3C 00F0          A3 AND LIT = B         % ISOLATE "M" FIELD         07346F00 D
167B  4809 0C41 0B0C 00F0          B L = B                % "M" INTO MS WORD OF B      07347E0C D
167C  4809 CC40 009C C0F0          A2 + B = MIR           % PERFORM ADDITION          07348C0C D
167D  78C9 00C0 0000 C0F0          IF AOV THEN SET LC1                                07349C00 D
167E  1E7C 0000 0000 C060          CARRY - 1 = CPCR                                   07350C00 D
167F  4F10 0000 0C9C 0060          CHECKOV - 1 = CPCR                                 07351C00 D
1680  4809 00C0 0D30 00F0          BMI                                                07352C00 D
1681  4809 0C40 8B9C 00FC          B R = MIR,B                                        07353C00 D
1682  0C00 00C0 0C9C 0020          16 = SAR                                           07354C00 D
1683  2000 00C0 0030 006C          SETCCA - 1 = CPCR      % SET THE CONDITION BITS    07355C00 D
1684  2F5C C000 000C C060          EOUTPUT - 1 = CPCR     % (R(A))+M INTO R(A)        07356C00 D
1685  66EC 0000 C00C 004C          OPCODE - 1 = MPCR                                  07357C0C D
                                                        %                           07358C00 D
                                                        %                           07359C00 D
                                   OP623:               % TYPE RL LITERAL ADD DOUBLE 07360C00 D
                                                        % (R(A),R(A+1)) + M INTO R(A),R(A+1)  07361C00 D
                                                        % SET CC, CARRY, AND OV BITS 07362C00 D
1686  4809 0C40 8B30 00F0          B R = B                                            07363C00 D
1687  9CFC 00C0 0C90 0080          4 = SAR; 15 = LIT                                  07364C0C D
1688  4809 E156 009C 00FC          A3 AND LIT = MIR       % "M" INTO MIR              07365C00 D
1689  4809 2C55 0B3C 0CF0          LIT AND B = B          % ISOLATE "A" FIELD         07366C0C D
168A  51C0 00F0 0C00 C060          REGSTACK - 1 = CPCR    % R(A) INTO MAR2            07367C00 D
168B  4809 CF01 0C30 C0FC          BMAR L = BR1           % TEMP STORAGE              07368C00 D
168C  000C C0C0 0030 C03C          COMP B = SAR                                       07369C0C D
168D  2F3C 00C3 0C0C C06C          EINPUT - 1 = CPCR      % (R(A)) INTO B             07370C30 D
168E  4809 CC41 2C3C C0F0          B L = A2               % (R(A)) INTO MS WORD OF A2  07371C0C D
168F  001C C0C3 009C 00A0          COMP 16 = SAR + 1 = LIT                            07372C00 D
1690  4809 2F5C 001C C0FC          LIT OR BMAR = MAR2     % R(A+1)                    07373CC0 D
1691  2F3C C0C0 0C9C 0060          EINPUT - 1 = CPCR      % (R(A+1)) INTO B           07374C0C D
1692  4809 CC5C 2D90 C0F0          A2 OR B = A2, BMI      % A2 = (R(A))/(R(A+1))       07375C0C D
1693  4809 CC40 0C90 C0FC          A2 + B = MIR                                       07376C00 D
1694  78C9 0000 0C0C 00F0          IF AOV THEN SET LC1                                07377C00 D
```

```
1695  1E70 0000 0000 0060          CARRY - 1 = CPCR                                  07378C00 D
1696  4F10 0000 0000 0060          CHECKOV - 1 = CPCR                                07379C00 D
1697  4909 C000 0000 00F0          BMI; SET LC1                                      0738CC0C D
1698  2000 0000 0000 C060          SETCCA - 1 = CPCR                                 07381C0C D
1699  4809 C000 0000 C0F0          BMI                                               07382C00 D
169A  4809 0C41 2000 C0F0          B L = A2                                          07383C00 D
169B  0000 0000 0000 C020          COMP 16 = SAR                                     07384C0C D
169C  4809 CC40 8000 00F0          B R = B                                           07385C00 D
169D  4809 C000 8C90 00F0          A2 R = MIR                 % NEW (R(A+1))         07386C00 D
169E  2F50 0000 0C00 C060          EOUTPUT - 1 = CPCR                                07387C0C D
169F  4809 0000 0C0C 20F0          ASR                        % REF ER1              07388C00 D
16A0  4809 0F40 8C1C 00FC          BMAR R = MAR2              % R(A)                 0738FC0C D
16A1  0000 C000 CC0C 0010          B = SAR                                           07390C00 0
16A2  4809 0C49 0090 00F0          B = MIR                    % NEW (R(A))           07391C00 0
16A3  2F50 0000 0000 C060          EOUTPUT - 1 = CPCR                                07392C00 D
16A4  66EC 0000 0000 C040          OPCODE - 1 = MPCR                                 07393C00 D
                                                              %                      07394C00 D
                                                              %                      07395C00 D
16A5  5F90 C000 0000 C060    OPCODE63:  XFCODE - 1 = CPCR     % "F" INTO A2          07396C00 D
16A6  4809 C640 0C40 00FC          A2 + AMPCR = AMPCR                                07397CC0 D
16A7  202C 0000 0F0C 00C0          OP63F - 1 = AMPCR                                 07398F0C D
16A8  4809 0900 0000 C0FC          STEP                                              07399C00 D
16A9  4824 00C0 0070 00F0          EXEC                                              0740CC00 D
                                                              %                      07401C00 0
16AA  2030 0000 0000 004C    OP63F:  OP630 - 1 = MPCR                                07402C0C D
16AB  2040 0000 0000 C040          OP631 - 1 = MPCR                                  07403C00 D
16AC  205C 00C0 0C00 0040          OP632 - 1 = MPCR                                  0740400C D
16AD  2060 0000 0000 C040          OP633 - 1 = MPCR                                  0740500C D
                                                              %                      07406C00 D
                                                              %                      07407000 D
                                   OP630:                     % TYPE RL LITERAL LOAD, M INTO R(A)  07408C00 D
                                                              % SET C*, CARRY AND OV BITS          0740900C 0
16AE  4809 2C56 0B0C C0F0          LIT AND B = B              % ISOLATE "M"           0741CC00 D
16AF  80FC 0000 0000 C080          15 = LIT; 4 = SAR                                 07411C00 D
16B0  4809 0C40 0030 00FC          B = MIR                                           07412C0C D
16B1  4809 E0C0 8B0C C0F0          A3 R = B                                          07413C00 D
16B2  4809 2C56 0E30 00FC          LIT AND B = B              % ISOLATE "A"          07414CC0 D
16B3  51CC 0000 0C0C 0060          REGSTACK - 1 = CPCR                               07415C00 D
16B4  2F50 C000 0020 C06C          EOUTPUT - 1 = CPCR         % WRITE "M" INTO R(A)  0741CC00 D
16B5  21B0 0000 0C00 C06C          SET00 - 1 = CPCR                                  07417C00 D
16B6  66EC 00C0 0C00 C040          OPCODE - 1 = MPCR                                 07418C00 D
                                                              %                      07419C0C D
                                                              %                      07420C0C D
                                   OP631:                     % FL TYPE LITERAL COMPARE, (R(A)):M  07421C00 D
                                                              % SET CC, CARRY, AND OV BITS         07422C00 D
1697  4809 2C56 0030 00FC          LIT AND B = MIR            % "M" INTO MIR          07423CC0 D
1698  00FC 00C0 0000 00EC          15 = LIT                                          07424000 D
1699  4809 E000 0B0C C0FC          A3 = B                                            07425C00 D
169A  228C 00C0 0C0C C060          CONTENTSRA - 1 = CPCR      % (R(A)) INTO B        07426C00 D
169B  4809 0C41 2000 00FC          B L = A2, BMI              % (R(A)) INTO MS WORD OF A2  07427C00 D
169C  0000 C0C0 0000 0020          COMP 16 = SAR                                     07428C0C D
169D  4809 0C41 0B00 00FC          B L = B                    % "M" INTO MS WORD OF B  0742CC00 D
169E  20C0 0000 CC30 0060          SETCC - 1 = CPCR           % SET THE CONDITION BITS  07430C00 D
169F  9849 CC5E 0C3D C0F0          A2 - B = MIR; SET LC2                             07431C00 D
16C0  7BC9 C000 00DC C0F0          IF AOV THEN SET LC1                               07432C00 D
16C1  1E70 0003 0000 C060          CARRY - 1 = CPCR                                  07433C0C D
16C2  4F1C 0003 0000 C060          CHECKOV - 1 = CPCR                                07434C00 D
16C3  66EC C0C0 C030 004C          OPCODE - 1 = MPCR                                 07435C00 D
                                                              %                      07436C00 D
                                                              %                      07437CC0 D
```

```
                                OP632:                                  % TYPE RL LITERAL MULTIPLY                                07438C00 D
                                                                        % R(A+1) X M = (R(A),R(A+1)), SET CC                      07439C00 D
                                                                                                                                  07440C00 D
16C4  2809 0C40 8B3C C0FC               B R = B; IF LC1                                                                           07441C00 D
16C5  8CFC 00C0 C030 0080               4 = SAR; 15 = LIT                                                                         07442C00 D
16C6  4809 2C56 C000 C0F0               LIT AND B = B                   % ISOLATE "A"                                             07443C00 D
16C7  51C0 00C0 0000 C060               REGSTACK - 1 = CPCR                                                                       07444C00 D
16C8  4809 0F41 0020 00FC               BMAR L = BR1                    % TEMP STORAGE                                            07445C00 D
16C9  0000 0000 0030 0030               COMP B = SAR                                                                              07446C00 D
16CA  4809 0F46 0B3C C0FC               BMAR + 1 = B                    % R(A+1)                                                  07447C00 D
16CB  51C0 00C0 0030 0060               REGSTACK - 1 = CPCR             % R(A+1) INTO MAR2                                        07448C00 D
16CC  2F30 00C0 0030 C060               EINPUT - 1 = CPCR               % (R(A+1)) INTO B                                         07449C00 D
16CD  4839 E156 2030 00F0               A3 AND LIT = A2                 % "M" INTO A2                                             07450C00 D
16CE  60FC 0003 0000 0060               15 = LIT                                                                                  07451C00 D
16CF  9C60 0000 003C C060               MULT - 1 = CPCR                 % (R(A+1)) X M INTO A3                                    07452C00 D
16D0  4809 E000 0B30 00F0               A3 = B, MIR                     % PRODUCT INTO B, MIR                                     07453C00 D
16D1  2000 00C0 0000 C060               SETCCA - 1 = CPCR               % SET THE CONDITION BITS                                  07454C00 D
16D2  4809 E001 0B30 00F0               A3 L = B                                                                                  07455C00 D
16D3  00CC 0000 C03C 0020               COMP 16 = SAR                                                                             07456C00 D
16D4  4809 CC40 8C30 CCF0               B R = MIR                       % (R(A+1))                                               07457C00 D
16D5  2F50 00C0 0030 0C60               EOUTPUT - 1 = CPCR                                                                        07458C00 D
16D6  4809 0000 0C00 20F0               ASR                             % REF FR1                                                 07459C00 D
16D7  4809 0F40 801C 00F0               BMAR R = MAR2                   % R(A)                                                    07460C00 D
16D8  0000 00C0 0C00 0010               8 = SAR                                                                                   07461C00 D
16D9  4809 E000 8030 00F0               A3 R = MIR                      % (R(A))                                                  07462C00 D
16DA  0000 0003 0030 0020               16 = SAR                                                                                  07463C00 D
16DB  2F5C 00C0 000F 006C               EOUTPUT - 1 = CPCR                                                                        07464C00 D
16DC  66E0 0003 0030 C040               OPCODE - 1 = MPCR                                                                         07465C00 D
                                                                        %                                                         07466C00 D
                                OP633:                                  % TYPE RL LITERAL DIVIDE                                  07467C00 D
                                                                        % (R(A),R(A+1))/M = R(A+1), REMAINDER                     07468C00 D
                                                                        % INTO R(A), SET CC AND OV                                07469C00 D
16D0  4809 2C56 0B00 00F0               LIT AND 0 = B                   % ISOLATE "M"                                             07470000 D
16DE  00FC 00C0 003C 00AC               15 = LIT; COMP 16 = SAR                                                                   07471C00 D
16DF  48C9 0C41 0C30 00F0               B L = MIR                       % LEFT JUSTIFY DIVISOR                                    07472C00 D
16EC  4809 E0C0 8B00 00FF               A3 R = B                                                                                  07473C00 D
16E1  30DC 00D0 003D 00C0               4 = SAR                                                                                   07474C00 D
16E2  4809 2C56 CB30 C0F0               LIT AND 0 = B                   % ISOLATE "A"                                             07475C00 D
16E3  51C0 00C0 0000 C06C               REGSTACK - 1 = CPCR             % R(A) INTO MAR2                                          07476C00 D
16E4  4809 0F41 0C20 00F0               BMAR L = BR1                    % STORE R(A)                                              07477C00 D
16E5  0000 00C0 003C 0030               COMP 8 = SAR                                                                              07478C00 D
16E6  2F3C 00C0 0000 006C               EINPUT - 1 = CPCR               % (R(A)) INTO B                                           07479C00 D
16E7  4809 CC41 20DC 00F0               B L = A2                                                                                  07480C00 D
16E8  00CC 00C0 0000 0C2C               COMP 16 = SAR                                                                             07481C00 D
16E9  4809 0F46 003C C0FC               BMAR + 1 = B                    % R(A+1)                                                  07482C00 D
16EA  51CC 00C0 000C 0060               REGSTACK - 1 = CPCR             % R(A+1) INTO MAR2                                        07483C00 D
16EB  2F30 00C0 0000 006C               EINPUT - 1 = CPCR               % (R(A+1)) INTO B                                         07484C00 D
16EC  49C9 CC5C 2030 00F0               A2 OR B = A2,BMI; SET LC1  % A2 = (R(A))/(R(A+1))                                         07485C00 D
16ED  2400 00C3 0C00 0060               DIV - 1 = CPCR                  % (R(A),R(A+1))/M                                         07486C00 D
16EE  4809 0000 003C 20F0               ASR                             % REF BR1                                                 07487C00 D
16EF  4809 0F40 801C 00F0               BMAR R = MAR2                   % R(A)                                                    07488C00 D
16F0  0000 00C3 0030 0010               8 = SAR                                                                                   07489C00 D
16F1  4809 C000 0030 00FC               A2 = MIR                        % REMAINDER                                               07490C00 D
16F2  2F50 00C0 0000 006C               EOUTPUT - 1 = CPCR                                                                        07491C00 D
16F3  4809 E0D0 0B3C 00FC               A3 = B, MIR                     % QUOTIENT INTO B, MIR                                    07492C00 D
16F4  200C 0000 0C0C C060               SETCCA - 1 = CPCR               % SET THE CONDITION CODE                                  07493C00 D
16F5  4809 0F46 0B3C 00F0               BMAR + 1 = B                    % R(A+1)                                                  07494C00 D
16F6  51C0 0C00 0030 C060               REGSTACK - 1 = CPCR             % R(A+1) INTO MAR2                                        07495C00 D
16F7  2F50 C0D0 C03C C06C               EOUTPUT - 1 = CPCR              % QUOTIENT INTO R(A+1)                                    07496C00 D
16F8  5050 0CC0 C00C C06C               CLEAROV - 1 = CPCR              % SET OV BIT                                              07497C00 D
```

```
16F9  66E0 0000 0000 0040            OPCODE - 1 = MPCR
                                                             *
                                                             *
16FA  5F90 0000 0000 0060   OPCODE64:  XFCODE - 1 = CPCR    * "F" INTO A2
16FB  4809 C640 0040 C0F0            A2 + AMPCR = AMPCR
16FC  2070 0000 0000 00C0            OP64F - 1 = AMPCR
16FD  4809 0000 0000 00F0            STEP
16FE  4824 0000 0000 C0F0            EXEC
                                                             *
16FF  3000 0000 0000 0040   OP64F:   FAULT - 1 = MPCR
1700  3000 0000 0000 0040            FAULT - 1 = MPCR
1701  3000 0000 0000 0040            FAULT - 1 = MPCR
1702  2080 0000 0000 0040            OP643 - 1 = MPCR
                                                             *
                                                             *
                            OP643:                           * RX TYPE BYTE SUBTRACT
                                                             * (R(A)) - (Y) BYTE INTO R(A)
                                                             * SET CC, CARRY, AND OV BITS
1703  4849 C000 C000 C0F0            SET LC2                 * BYTE OPERATION FLAG FOR RXMFIELD
1704  5700 0000 0000 C060            RXMFIELD - 1 = CPCR     * Y INTO B
1705  4809 0C41 0C10 C0F0            B L = BR2
1706  000C 0000 0000 0030            COMP 8 = SAR
1707  60E0 0000 0000 C060            EMULIN - 1 = CPCR       * (Y) INTO B
1708  4809 0C41 0030 00FC            B L = MIR
1709  0000 C000 0030 0020            COMP 16 = SAR
170A  4809 EC00 0B00 C0FC            A3 = B
170B  2260 0000 C000 006C            CONTENTSRA - 1 = CPCR   * (R(A)) INTO B
170C  4809 CC41 2000 00FC            B L = A2,BMI            * (R(A)) INTO MS WORD OF A2
170D  000C C000 0000 0020            COMP 16 = SAR
170E  380B C000 0C00 00F0            IF LC2 THEN STEP ELSE SKIP
170F  2090 0000 0C0C C04C            LS643 - 1 = MPCR
1710  4809 0C40 8B0C 00F0            B R = B                 * GET MS BYTE OF (Y)
1711  000C 0000 0000 C030            24 = SAR
1712  4809 CC41 0B00 00F0            B L = B
1713  000C C000 0000 C020            COMP 16 = SAR
1714  4849 CC5E 0C80 00F0   C643:    A2 - B = MIR; SET LC2
1715  7BC9 0000 0C0C 00F0            IF AOV THEN SET LC1
1716  1E70 C000 C000 C060            CARRY - 1 = CPCR
1717  4F1C 0000 CC00 0060            CHECKOV - 1 = CPCR
1718  4809 0000 0D00 0CF0            BMI
1719  4809 0C40 8B3C 00F0            B R = B, MIR
171A  00CC 0000 0C0C 0020            16 = SAR
171B  2F5C 0000 0000 C060            EOUTPUT - 1 = CPCR      * WRITE NEW (R(A))
171C  2000 00C0 0000 C060            SETCCA - 1 = CPCR
171D  66EC 0000 0000 0040            OPCODE - 1 = MPCR
171E  4809 0C41 0B00 40F0   LS643:   B L = B, CSAR
171F  0000 C000 0000 CC30            COMP 8 = SAR
1720  4809 0C40 8B00 C0FC            B R = B                 * LS BYTE ISOLATED INTO B
1721  7130 0000 C00C 005C            C643 - 1 = MPCR
                                                             *
                                                             *
1722  5F90 0000 000C 0060   OPCODE65:  XFCODE - 1 = CPCR    * "F" INTO A2
1723  4809 C640 0C40 C0F0            A2 + AMPCR = AMPCR
1724  20AC CC00 C000 C0C0            OP65F - 1 = AMPCR
1725  4809 C000 CC0C C0FC            STEP
1726  4824 C000 0C0C C0F0            EXEC
                                                             *
1727  3000 0000 0000 C040   OP65F:   FAULT - 1 = MPCR
1728  300C 0000 0C00 0040            FAULT - 1 = MPCR
1729  300C 0000 0C0C C040            FAULT - 1 = MPCR
```

```
172A  2080 00C0 0000 C040          OP653 - 1 = MPCR                                    07558C00 D
                                                  %                                    07559C00 C
                                                  %                                    0756CC00 D
                                   OP653:         % TYPE RX BYTE ADD                   07561C00 D
                                                  % (R(A)) + (Y) BYTE INTO R(A)        07562C00 D
                                                  % SET CC, CARRY, AND OV BITS         0756JC0C D
172B  4849 0003 0030 00F0          SET LC2        % RXMFIELD BYTE FLAG                 0756 4C00 D
172C  5700 0000 0000 0060          RXMFIELD - 1 = CPCR  % Y INTO B                     07565C0C D
172D  4809 0C41 0010 00F0          B L = BR2                                           07566C00 D
172E  0000 00C3 0C30 0030          COMP 8 = SAR                                        07567C00 D
172F  50EC 00C0 0030 C060          EMULIN - 1 = CPCR                                   0756BC00 D
1730  4809 0C41 0C30 C0F0          B L = MIR                                           07569C00 D
1731  0000 00C0 0070 0020          COMP 16 = SAR                                       07570C00 D
1732  4809 E0C0 0B30 00F0          A3 = B                                              07571CC0 D
1733  22B0 00C3 0030 0060          CONTENTSRA - 1 = CPCR % (R(A)) INTO E               07572C0C D
1734  4809 0C41 2020 C0F0          B L = A2, BMI                                       07573C00 D
1735  0000 0003 0070 0020          COMP 16 = SAR                                       07574C00 D
1736  380B CCC0 0070 00F0          IF LC2 THEN STEP ELSE SKIP                          07575C0C D
1737  20CC 00C0 0000 0040          LS653 - 1 = MPCR                                    07576C00 D
1738  4809 0C40 8B0C C0F0          B R = B              % ISOLATE MS BYTE OF (Y)        07577C00 E
1739  0000 0003 0020 003C          24 = SAR                                            07578C00 D
173A  4809 0C41 0B0C C0F0          B L = B                                             07579C0C D
173B  0000 00C3 0000 0020          COMP 16 = SAR                                       0758CC00 D
173C  1809 0C40 0C30 C0F0  C653:   A2 + B = MIR         % PERFORM ADDITION           07581000 D
173D  78C9 00C0 0030 00F0          IF ADV THEN SET LC1                                 07582C00 D
173E  1E70 00C0 0000 0060          CARRY - 1 = CPCR                                    07583C06 D
173F  3F10 0003 0020 0060          CHECKOV - 1 = CPCR                                  07584C00 E
1740  4809 0000 C0D0 0CF0          BMI                                                 07585CC0 D
1741  4809 0C40 8B3C 00F0          B R = B, MIR                                        07586C00 D
1742  0000 CCC0 000C 0020          16 = SAR                                            07587C00 D
1743  2F50 C000 0000 C06C          EOUTPUT - 1 = CPCR                                  07588C00 D
1744  20DC 00C0 0030 C060          SETCCA - 1 = CPCR    % SET THE CONDITION BITS       07589C00 D
1745  56EC 00C0 000C C040          OPCODE - 1 = MPCR                                   0759CC00 D
1746  48C9 0C41 0B00 40F0  LS653:  B L = B, CSAR        % GET LS BYTE OF (Y)           07591CC0 D
1747  0C0C C0C0 0030 C030          COMP 8 = SAR                                        07592C00 D
1748  4809 0C40 8B3C 00F0          B R = B                                             07593C00 D
1749  73BC 00C0 0000 C050          C653 - 1 = MPCR                                     07594C00 D
                                                  %                                    07595C00 D
                                                  %                                    07596C00 D
174A  5F9C 00C3 0030 0060  OPCODE66:  XFCODE - 1 = CPCR                               07597C06 D
174B  4809 C640 004C C0F0          A2 + AMPCR = AMPCR                                  07598CC0 D
174C  20D0 00C0 003C C0C0          OP66F - 1 = AMPCR                                   07599CC0 D
174D  4809 C000 0020 00F0          STEP                                                0760CC00 D
174E  4824 CCC0 0030 00F0          EXEC                                                07601C0C D
                                                  %                                    07602C00 D
174F  360C C0C0 0C00 0040  OP66F:  FAULT - 1 = MPCR                                   0760300C D
1750  300C C0C0 0C00 0040          FAULT - 1 = MPCR                                    07604C0C D
1751  300C C0C3 0030 0040          FAULT - 1 = MPCR                                    07605C00 D
1752  20E0 00C0 000C C04C          OP663 - 1 = MPCR                                    07606C0C D
                                                  %                                    07607C0C D
                                   OP663:         % RX TYPE BYTE COMPARE, (R(A)):(Y) BYTE  07608C0C D
                                                  % SET CC, CARRY, AND OV BITS         07609C00 D
1753  4849 0000 CC00 00F0          SET LC2        % BYTE OPERATION FLAG FOR EXMFIELD   07610C00 D
1754  5700 0000 0C00 C060          RXMFIELD - 1 = CPCR  % Y INTO B                     07611C06 D
1755  4809 0C41 0C10 00F0          B L = BR2                                           07612C00 D
1756  0C0C 0CC0 0030 003C          COMP 8 = SAR                                        07613C00 D
1757  50EC 00C3 000C 0060          EMULIN - 1 = CPCR    % (Y) INTO B                   07614C00 D
1758  4809 0C41 C030 CCFC          B L = MIR                                           07615C0C D
1759  0000 00C0 C03C 0020          COMP 16 = SAR                                       07616C00 D
175A  4809 E0C0 0B30 C0F0          A3 = B                                              07617CC0 D
```

```
175B  2280 0C00 0000 006C            CONTENTSRA - 1 = CPCR                                  07616C00 D
175C  4809 0C41 2D30 00F0            B L = A2, BMI                                          07617000 D
175D  0000 0000 0000 002C            COMP 16 = SAR                                          07620000 D
175E  3808 0000 0000 00F0            IF LC2 THEN STEP ELSE SKIP                             07621000 D
175F  2CFC CC00 0000 0040            LS663 - 1 = MPCR       % GET LS BYTE OF (Y)            07622C00 D
1760  4809 0C40 8B0C CCF0            B R = B                                                07623000 D
1761  0000 C0C0 0030 0030            24 = SAR                                               07624C00 D
1762  4809 0C41 1B3C 00FC            B L = B, A3            % MS (Y) BYTE INTO MSW OF E, A3  07625000 D
1763  0000 0003 0C00 C020            COMP 16 = SAR                                          07626000 D
1764  4849 CC5E 0030 C0F0  C663:     A2 - B = MIR; SET LC2                                  07627C00 D
1765  7BC9 00C0 0000 C0F0            IF AOV THEN SET LC1                                    07628CC0 D
1766  1E70 0003 0C00 0060            CARRY - 1 = CPCR                                       07629000 D
1767  4F1C 0000 0000 0060            CHECKOV - 1 = CPCR                                     07630000 D
1768  4809 E0C0 0B0C 00FC            A3 = B                 % RESTORE (Y) INTO B            07631000 D
1769  2C00 0003 0C30 0060            SETCC - 1 = CPCR       % SET THE CONDITION BITS        07632C00 D
176A  66EC 0000 CC30 0040            OPCODE - 1 = MPCR                                      07633C0C D
176B  4809 0C41 0B3C 40F0  LS663:    B L = B, CSAR                                          07634000 D
176C  0000 0000 000C C03C            COMP 8 = SAR                                           07635C0C D
176D  4809 0C40 9B3C 00F0            B R = B, A3                                            07636CC0 D
176E  763C 0000 000C 0050            C663 - 1 = MPCR                                        07637CCC D
                                     %                                                      07638000 D
                                     %                                                      07639C00 D
176F  5F90 00C0 0C3C C060  OPCODE67: XFCODE - 1 = CPCR                                      07640C00 D
1770  4809 C640 0C40 00FC            A2 + AMPCR = AMPCR                                     07641000 D
1771  2100 0003 0030 C0C0            OP67F - 1 = AMPCR                                      07642C00 D
1772  4809 0000 000C 00FC            STEP                                                   07643CC0 D
1773  4824 00C0 0C3C C0F0            EXEC                                                   07644000 D
                                     %                                                      07645CC0 D
1774  4E5C 0000 0000 004C  OP67F:    NOTIMP - 1 = MPCR                                      07646C00 D
1775  4E50 0000 000C C04C            NOTIMP - 1 = MPCR                                      07647C00 D
1776  4E50 0000 0000 0040            NOTIMP - 1 = MPCR                                      07648CC0 D
1777  2110 0003 003C 0040            OP673 - 1 = MPCR                                       07649C00 D
                                     %                                                      07650C00 D
                                     %                                                      07651C0C D
                               OP673:                % TYPE RX BYTE COMPARE AND INDEX BY 1 07652C00 D
                                                     % (R(A))+(Y) BYTE, (R(M))+1 INTO R(M)  07653000 D
                                                     % SET CC, CARRY, AND OV BIT            07654C0C D
1778  4849 0000 000C 00F0            SET LC2                % BYTE OPERATION FLAG FOR RXMFIELD 07655C00 D
1779  570C 0000 0C00 C060            RXMFIELD - 1 = CPCR    % Y INTO B                      07656C00 D
177A  4809 0C41 0D1C C0FC            B L = BR2                                              07657000 D
177B  0C0C 0000 0000 0030            COMP 8 = SAR                                           07658CC0 D
177C  60E0 0C00 000C 0060            EMULIN - 1 = CPCR      % (Y) INTO B                    07659C00 D
177D  4809 0C41 0030 C0F0            B L = MIR                                              07660C00 D
177E  000C 00C0 000C 0020            COMP 16 = SAR                                          07661C00 D
177F  4809 E003 0B00 00F0            A3 = B                                                 07662C00 D
1780  2280 0000 000C 0060            CONTENTSRA - 1 = CPCR  % (R(A)) INTO B                 07663C00 D
1781  4809 0C41 2D30 0CF0            B L = A2, BMI          % (R(A)) INTO MS WORD OF A2     07664C0C D
1782  000C 00C3 0C00 002C            COMP 16 = SAR                                          07665C00 D
1783  38CB C0C0 0030 00F0            IF LC2 THEN STEP ELSE SKIP                             07666C00 D
1784  212C 0003 0030 004C            LS673 - 1 = MPCR                                       07667CC0 D
1785  4809 CC40 9B00 00F0            B R = B                                                07668C00 D
1786  0000 0000 0C30 003C            24 = SAR                                               07669C00 D
1787  4809 0C41 1B0C 00F0            B L = B, A3            % MS (Y) BYTE INTO MS WORD OF B 07670C00 D
1788  0000 0003 C000 C02C            COMP 16 = SAR                                          07671C00 D
1789  4849 CC5E 0030 00F0  C673:     A2 - B = MIR; SET LC2                                  07672C00 D
178A  7BC9 0CC0 0030 C0FC            IF AOV THEN SET LC1                                    07673C00 D
178B  1E70 C00D 0C3C 0060            CARRY - 1 = CPCR                                       07674C00 D
178C  4F1C 00C0 000C 0060            CHECKOV - 1 = CPCR                                     07675CCC D
178D  4809 E0C3 0B3C C0F0            A3 = B                                                 07676C0C D
178E  2006 00C3 0C00 C06C            SETCC - 1 = CPCR                                       07677CCC D
```

```
178F  2380 6000 0000 0060          CONTENTSRM - 1 = CPCR  % (R(P)) INTO B          07678000 0
1790  4809 0C46 0C30 00F0          B + 1 = MIR            % (R(M)) + 1             07679000 0
1791  2F5C 0003 003C 0060          EOUTPUT - 1 = CPCR                              07680000 0
1792  66E0 0003 0000 0040          OPCODE - 1 = MPCR                               07681000 0
1793  4809 0C41 0B0C 40F0  LS673:  B L = B, CSAR          % ISOLATE LS BYTE OF (Y)  07682000 0
1794  000C 00C0 CC0C 0030          COMP B = SAR                                    07683000 0
1795  4809 0C40 9B3C 00F0          B R = B, A3                                     07684000 0
1796  788C 0003 0C0C 0050          C673 - 1 = MPCR                                 07685000 0
                                                                                  07686000 0
                                                                                  07687000 0
                                   OPCODE70:              % NOT IMPLEMENTED         07688000 0
1797  4E5C 0000 0000 C04C          NOTIMP - 1 = MPCR                               07689000 0
                                                                                  07690000 0
                                   OPCODE71:              % NOT IMPLEMENTED         07691000 0
1798  4E5C 0003 0000 0040          NOTIMP - 1 = MPCR                               07692000 0
                                                                                  07693000 0
                                   OPCODE72:              % NOT IMPLEMENTED         07694000 0
1799  4E50 0000 0C30 C040          NOTIMP - 1 = MPCR                               07695000 0
                                                                                  07696000 0
                                   OPCODE73:              % NOT IMPLEMENTED         07697000 0
179A  4E50 0000 0030 0040          NOTIMP - 1 = MPCR                               07698000 0
                                                                                  07699000 0
                                   OPCODE74:              % NOT IMPLEMENTED         07700000 0
179B  4E50 0000 000C 0040          NOTIMP - 1 = MPCR                               07701000 0
                                                                                  07702000 0
                                   OPCODE75:              % NOT IMPLEMENTED         07703000 0
179C  4E50 0000 003C C040          NOTIMP - 1 = MPCR                               07704000 0
                                                                                  07705000 0
                                   OPCODE76:              % NOT IMPLEMENTED         07706000 0
179D  4E50 0000 0000 C040          NOTIMP - 1 = MPCR                               07707000 0
                                                                                  07708000 0
                                   OPCODE77:              % NOT ASSIGNED            07709000 0
179E  3000 0000 0000 C040          FAULT - 1 = MPCR                                07710000 0
                                                                                  07711000 0
179F  0000 0000 0000 C040          END                                             07712000 0
1784  7920 0000 0000 0050  ?END
1777  7770 0000 0000 C050
1771  7730 0003 0000 C00C
175F  76A0 0000 0000 0050
1752  7520 0003 0000 0050
174C  74E0 0000 000C 00DC
1737  7450 0003 0000 C050
172A  72A0 0000 00DC 0050
1724  7260 0000 0030 0000
170F  7100 0003 0030 0050
1702  7020 0003 0030 0050
16FC  6FE0 0000 0000 C00C
16AD  5DC0 0003 0030 0050
16AC  6C30 0003 0000 0050
16AB  5B60 0003 0000 0050
16AA  6A0C 0000 0030 0050
16A7  6A90 0003 000C 0000
1648  6850 0003 0000 0050
1647  5760 0003 0000 0050
1646  6570 0000 0030 0050
1645  6480 0000 0000 0050
1642  6440 0003 0000 0000
15DA  6230 0003 0000 0050
15D9  6020 0003 0000 0050
15D8  5EE0 0003 0000 0050
```

```
15D7  5DA0 0000 0C00 0C5C
15D4  5D6C 00C0 0C0C 0000
1573  5AAC 0000 000C 0050
1572  58FC 0000 0000 C050
1571  57BC 0000 C00C C050
1570  573C 00C0 C000 0050
156D  56FC 0000 0C00 000C
152D  552C C0C0 0000 C050
152B  53E0 0C03 000C C05C
152A  52DC 0C03 0C0C 0050
1527  529C 0003 0C0C 0000
14F5  50FC 0C03 0000 C050
14F3  506C 0FC0 0000 0050
14F2  4F5C 00C0 000C C05C
14EF  4F1C C003 0C30 0000
14B2  4D9C 0000 0000 0050
14B1  4C50 00C0 C000 C050
14B0  4BE0 0000 0000 005C
14AF  4B2C 0CC0 0070 C050
14AC  4AEC 0000 0030 C000
1470  495C 00C0 CC3C 0050
146F  4820 0CC3 003C C050
146E  47CC 0C00 0C00 C05C
146D  470C 0C00 0C3C C050
146A  46C0 0C03 000C 0000
1436  456C 0CC0 0000 005C
1435  4450 00C3 0C00 C05C
1434  4400 C0C0 0030 0050
1433  436C 00C0 0C30 005C
1430  4320 00C3 0F0C C00C
13FF  41FC 0000 0C00 C050
13FE  40EC 0003 C00C 0050
13FD  4090 CC00 C000 0050
13FC  3FFC 0C00 0C00 005C
13F9  3FBC 00C3 000C 0000
13B2  3E60 C0C0 0C0C 0050
13B1  3D0C 0D00 003C 005C
13B0  3B2C 0C00 0C00 0050
13AC  3AEC 00C0 C000 C000
1371  396C 00C0 003C C050
1370  380C C0C0 0C30 C050
136E  371C 00C3 0C0C 0050
136B  36D0 0000 0C0C C000
1328  353C 0CC0 0030 C050
132A  33D0 C0C3 0C3C 005C
1329  338C 0003 0C0C 0050
1328  328C 00C0 C00C C050
1325  3270 00C0 0C30 C000
12FA  3180 0C00 0C0C 0050
12F9  305C 0CC0 0C00 C05C
12F7  2FAC C0C3 0C00 C05C
12F4  2F6C C0C0 000C C000
12F1  2F1C 00C3 0030 C05C
12EE  2F1C CCC0 000C C050
12E8  2F1C 0003 0C0C C05C
12E5  2F1C 00C0 0C0C CC50
12E0  2F1C 0CC3 0C0C C050
12DD  2F1C 0C03 000C 0050
12DC  2F1C 00C0 0C30 0050
12D7  2F1C 0903 003C C05C
```

```
1202  2F1C 0000 0C0C C050
120D  2F1C 0000 0030 0050
1208  2F10 0000 0C00 C050
12C4  2F1C 00C0 0000 0C50
12C0  2E80 00C0 0000 C05C
120F  2E00 0000 0030 0050
129F  2000 0000 0030 C050
1280  20C0 00C0 C030 0050
128A  2D7C 00C0 0000 0050
1289  2D2C 00C0 C000 0050
1280  2C00 00C0 0030 C050
1287  2C8C 00C0 0000 0050
1286  2C40 00C0 0030 0050
1285  2C00 00C0 0030 0050
1282  2B40 0000 0000 0000
12A9  3210 00C0 0000 0C50
1291  2990 0000 6000 0050
1264  26D0 0000 0030 005C
1244  2880 0000 0030 0050
1243  2780 0000 0030 005C
1242  2580 0000 0000 0050
1241  2440 C0C0 0030 005C
123E  2400 0000 0000 0000
11EB  2280 0000 0000 0050
11EA  20E0 0300 0030 0050
11E9  1FBC 00C0 0C0C 005C
11E8  1EBC 00C0 0030 0050
11E5  1E70 0000 0000 C0D0
117A  1C90 0000 0030 005D
1179  1A9C 0000 0000 0050
1178  1900 00C0 0030 005C
1177  17A0 0000 0070 0050
1174  1760 0000 0050 C0DC
1130  1650 00C0 0030 0050
113B  1520 00C0 0030 0050
113A  146C 0000 C000 0050
1139  13C0 0000 0000 005C
1136  1380 0000 CC00 C0DC
10FE  1270 C000 0020 0050
10F0  1140 00C0 0000 0050
10EC  1080 C0C0 0C0C 005C
10EB  0FEC 00C0 0000 0050
10E8  0FA0 0000 0000 0000
10C0  0E90 0000 C030 C050
10BF  0060 00C0 0C00 0050
10BE  0CAC 0000 0030 0050
10B0  0C00 C0C0 0030 C050
10BA  0BC0 0000 0000 C000
1017  0920 00C0 0000 005C
1016  0630 00C0 0000 C050
1015  03CC 0000 0000 005C
1014  0170 0000 0000 C050
1011  0130 00C0 0C0C C00C
0F8C  FFC0 C0C0 0070 C04C
0F88  FCB0 00C0 0000 C040
0F8A  FAA0 0000 0000 C040
0F89  F8C0 00C0 0000 0040
0F86  F880 00C0 0000 C0C0
0F30  F680 00C0 0030 C040
0F2E  F4B0 0000 0000 C040
```

```
0F2D  F300 0000 0000 0040
0F2A  F2C0 0000 0000 00C0
0FDE  F150 0000 0030 0040
0EDD  EFD0 C000 0000 0040
0EDC  EEC0 0000 0030 0040
0EDD  EDE0 0000 0030 0040
0ED8  EDA0 0000 0000 00C0
0E5E  EAD0 00C0 0000 0040
0E5C  E850 0000 0030 0040
0E58  E5E0 0000 0030 004C
0E58  E5A0 C003 0000 00C0
0E00  E410 0000 0030 0040
0DFF  E270 00C0 0C00 004C
0DFE  E120 0000 000C 004C
0DFD  E000 00C0 0030 0040
0DFA  DFC0 0000 0000 00C0
0D79  DCD0 0000 0030 0040
0D77  DA10 0000 000C 0040
0D76  D790 C000 0030 004C
0D73  D750 00C0 0000 00C0
0D19  D5C0 0000 0000 004C
0D18  D420 0000 0030 0040
0D17  D2B0 0000 0030 0040
0D16  D190 0000 0030 0040
0D13  D150 0000 0030 00CC
0CBF  D0A0 0000 0000 0040
0CBE  CE30 0003 0000 0040
0CBD  CD00 0000 0030 0040
0CBC  CBF0 0000 0030 0040
0CB9  CBB0 C000 0030 00CC
0C32  C9D0 0000 0000 0040
0C31  C710 00C3 0000 0040
0C30  C560 0000 0000 0040
0C2F  C320 0003 0000 0040
0C2C  C2E0 0000 0000 00C0
0BD5  C1D0 0000 0000 0040
0BD4  BF90 0000 0000 004C
0BD3  9EB0 0000 0000 0040
0BD2  BD50 0000 0030 0040
0BCF  BD10 00C0 0000 00CC
0BBC  BC60 0000 0030 0060
0B97  BA30 0000 0000 0040
0B6F  B7B0 0000 0030 0040
0F64  BAA0 0000 0030 0040
0F63  B820 0003 0030 0040
0B61  B640 0000 0030 0040
0B5E  B600 00C3 0000 00C0
0AE9  B380 00C3 0030 0040
0AE8  B0C0 C000 0000 004C
0AE6  AE90 0000 0000 0040
0AE3  AE50 00C0 0030 00CC
0A72  ACA0 0003 0000 0040
0A71  AA30 0000 0030 0040
0A70  A910 0000 0030 0040
0A6F  A720 C000 0030 0040
0A6C  A6E0 0000 0030 00C0
0A1F  A5D0 0000 0030 004C
0A1E  A300 C000 0000 0040
0A1D  A360 0000 0030 0040
0A1C  A1F0 0000 0000 0040
```

```
0A19  A19C 0000 0000 00C0
0A0D  A12D 0000 0000 0040
0A05  A0D0 00C0 0000 0040
09DD  9FBC 00C0 0000 0040
09DC  9E7C 0000 0000 0040
09DA  9D0C 0000 0030 0040
09D7  9D90 0000 0030 00C0
0998  99EC 0000 0030 0040
098D  9D2C 0000 0030 0040
0989  9A4C 0000 0030 0040
0988  9880 00C0 000C 0040
0985  987C 00C0 0000 00C0
0962  97FC 0000 0030 0040
0960  970C 0000 0000 0040
095F  9620 0000 0000 0040
095C  95EC 0000 0000 00C0
0934  9570 0000 0000 0040
0932  9420 0000 0030 0040
0931  9340 0000 0000 0040
092E  930C 0000 0030 00C0
0908  90F0 0000 000C 0040
0909  908C 00C0 0030 0040
0900  9CFC 0000 0000 0040
08FE  90F0 0000 0000 0040
08A7  8ABC 00C0 0030 0040
0898  8EC0 0000 0000 0040
0F8D  8DB0 0000 0000 0040
088C  9C8C 0000 0000 0040
0F9B  88EC 0000 0030 0040
0888  88A0 0000 0000 00C0
0880  916C 0000 0000 0040
0B7D  9B00 0000 0030 0040
0F7A  97CC 0000 0030 00C0
07FF  8420 0000 0000 0040
07FE  82CC 00C0 0000 0040
07FD  822C 0000 0000 0040
07FB  812C 00C0 0000 0040
07FA  809C 0000 0000 0040
07F9  7FF0 00C0 0030 0040
07F3  7F80 00C0 0000 00C0
07EF  951C 00C0 0030 0040
07EC  7EF0 00C0 0030 0040
07E9  7E80 0000 0000 00CC
0771  777C 00C0 000C 0040
0765  771C 0000 0000 0040
073E  7D0C 0000 0000 0040
073D  7C30 0000 0030 0040
073C  7B60 0000 0030 0040
0738  7A90 0000 000C 0040
0739  7A3C 00C0 0000 0040
0738  78BC 00C0 0000 0040
0737  77F0 0000 0000 0040
0735  75D0 0000 0030 0040
0734  74F0 0000 0000 0040
0733  73E0 0000 0030 0040
072D  7320 0000 0000 00C0
0724  7E3C 00C0 000C 0040
0722  7D0C 0000 0000 0040
0721  7240 0000 000C 0040
071E  720C 0000 0030 00C0
```

```
06DD  7CD0 0000 0000 C040
06DC  6F80 0000 0000 CC40
06DB  6E9C 00C0 0030 C040
06DA  6DDC 00C3 0030 0040
06D7  6D9C 00C0 0030 C0CC
06C4  6C40 0FC0 CC3C C040
06BE  6C00 00C0 0030 00C0
06BB  79DC 0000 C0C0 C050
06BA  79CC 00C3 0C30 C050
06B9  79B0 0CC0 0CCC 005C
06B8  79A0 0000 CCCC C05C
06B7  799C 00C0 0030 0050
06B6  798C 0CC0 0C0C C05C
06B5  797C 00C0 000C 0050
06B4  796C 0003 0030 005C
06B3  76EC 0FC0 C00C 0050
06B2  749C 00C0 0C00 C05C
06B1  7210 00C0 0030 C050
06B0  6F9C 0000 0C30 0050
06AF  6A40 00C0 0C00 C050
06AE  63F0 0000 00C0 005C
06AD  5D10 00F3 0CFC C050
06AC  56AC 00C3 000C 005C
06AB  569C C000 CC0C C05C
06AA  568C 00C0 C00C 0050
06A9  524C CC00 0000 0050
06A8  4EC0 00C0 0030 0050
06A7  4EBC 0CC0 000C C050
06A6  4EA0 0CC0 0030 005C
06A5  4E90 00C0 0000 005C
06A4  4E8C 0FC0 00CC 0050
06A3  4A9C 00C0 C00C C050
06A2  4670 CCC0 000C C05C
06A1  420C 0FC0 0000 0050
06A0  3F6C 00C0 0C00 005C
069F  3A9C 0003 0C30 0C50
069E  368C 00F3 C000 C05C
069D  3220 0FC0 C00C C05C
069C  2A50 0CC0 0030 005C
069B  2A4C 00C3 C00C 005C
069A  2A3C 00C0 0C3C C050
0699  23BC 00C3 00CC C050
0698  1E2C C0C0 0CCC C050
0697  1710 00C3 0C3C C05C
0696  133C 00C0 C00C 0050
0695  0F50 00C0 0CC0 0050
0694  0B7C 00C0 00CC C05C
0693  00E0 0000 0C3C C050
0692  F830 CCC0 00CC 0040
0691  F27C 0000 0C3C 004C
0690  ED50 00C0 0C30 C04C
068F  E55C 00C0 0C0C C040
068E  DF7C 00C0 0030 0C40
068D  D700 00C0 0C0C C040
068C  D100 00C0 C00C CC40
068B  CB6C 00C3 CC3C 004C
068A  C290 0CC0 00CC 0040
0689  BCCC 0C00 0000 004C
0688  B5BC 00C3 C03C 0C40
0687  AE0C C0C0 0C0C 0040
```

```
0686  A690 0000 0000 0040
0685  A160 0000 0000 0040
0684  7040 0000 0000 0040
0683  9820 0000 0000 0040
0682  9590 0000 0000 0040
0681  92B0 0000 0000 0040
0680  8770 0000 0000 0040
067F  7E60 0000 0000 0040
067E  7190 0000 0000 0040
067D  6040 0000 0000 0040
067C  6BB0 0000 0000 0040
0679  67B0 0000 0000 0000
0626  64F0 0000 0000 0040
0613  64F0 0000 0000 0040
060E  6180 0000 0000 0040
060D  62B0 0000 0000 0040
05D6  66E0 0000 0000 0040
0595  5B50 0000 0000 0040
05B0  5B50 0000 0000 0040
05AB  5B50 0000 0000 0040
05A6  5B50 0000 0000 0040
05A1  5B00 0000 0000 0040
05A0  5AB0 0000 0000 0040
059D  5A60 0000 0000 0040
059A  5A10 0000 0000 0040
0587  6330 0000 0000 0060
0582  6330 0000 0000 0060
0581  5860 0000 0000 0040
0580  5930 0000 0000 0040
057D  5860 0000 0000 0040
0579  5810 0000 0000 0040
0567  56E0 0000 0000 0040
0550  5330 0000 0000 0060
0522  5270 0000 0000 0040
051C  5D60 0000 0000 0040
0518  5300 0000 0000 0060
051A  5E50 0000 0000 0060
0518  5E50 0000 0000 0060
0511  5E50 0000 0000 0060
0505  5050 0000 0000 0040
0504  50A0 0000 0000 0040
0502  50A0 0000 0000 0040
0501  5050 0000 0000 0040
04FF  3050 0000 0000 0040
04FE  30A0 0000 0000 0040
04FC  4FF0 0000 0000 0040
04F8  5050 0000 0000 0040
04F1  50F0 0000 0000 0040
04EC  5B00 0000 0000 0060
04E8  50F0 0000 0000 0040
04E6  3B00 0000 0000 0060
04D3  4E20 0000 0000 0040
04CA  4DF0 0000 0000 0040
04AF  61E0 0000 0000 0060
04A6  50E0 0000 0000 0060
0496  51C0 0000 0000 0060
048F  60E0 0000 0000 0060
048A  60E0 0000 0000 0060
047B  47B0 0000 0000 0040
0475  60E0 0000 0000 0060
```

```
046F  46F0 0000 0030 0040
046D  51C0 0000 000C 0060
0469  46FC 0000 0000 0040
0467  51CC 000C 0000 0060
0464  46F0 0000 0030 0040
0462  51C0 0000 0030 0060
045F  47DC 0000 000C 0040
045D  50EC 000C 0000 0060
0458  469C 0000 0030 0040
0457  4640 0000 0000 0040
0456  95F0 000C 0000 0640
0455  9580 0000 0030 0040
0452  954C 0000 0030 000C
044A  50E0 0000 000C 006C
0444  4440 000C 000C 0040
0442  51C0 0000 0000 0060
043E  4440 000C 0000 0040
043C  444C 0000 000C 0040
043A  51C0 0000 0000 0060
0436  60EC 0000 0030 0060
0431  43EC 0000 0030 0040
0430  43C0 0000 0000 0040
042F  437C 000C 000C 0040
042E  4310 0000 0000 0040
042B  42D0 0000 0000 000C
0425  44C0 000C 000C 0040
0424  4250 000C 0030 0040
041F  60E0 0000 0000 006C
0416  416C 0000 000C 0040
0415  6330 0000 0030 0060
0410  416C 000C 0000 0040
0406  6330 000C 0030 006C
03F0  66E0 0000 0000 0040
03E3  3F70 0000 000C 0060
03D5  66E0 0000 0030 0040
03C8  3F70 0000 0000 0060
03BE  4E50 0000 000C 0040
03BD  3DCC 000C 0000 0040
03BC  3DE0 0000 0030 0040
03B9  3B80 0000 0030 0000
03AC  66E0 000C 000C 0040
03A2  3F70 0000 0000 0060
03AE  66E0 000C 0000 0040
0383  3F70 0000 0000 0060
0376  4E50 0000 0000 0040
0375  3990 0000 0000 0040
0374  3760 0000 0030 0040
0371  3730 0000 000C 0000
035E  3620 0000 0000 0040
0357  51E0 0000 0030 0060
0330  50E0 0000 0000 006C
0348  61E0 0000 0000 0060
0340  60E0 000C 0000 0060
0310  61EC 0000 0000 0060
0314  60E0 0000 0000 0060
030F  60E0 000C 0000 0060
0305  50FC 0000 0030 0040
0301  5BD0 0000 0000 0060
02F0  2F30 0000 0000 0060
02EC  2F30 0000 0000 0060
```

```
02E9  2F3C 0000 0000 0060
02E6  2F3C 0000 0000 0060
02E3  2F3C 0000 0000 0060
02E0  2F3C 0000 0000 0060
02DD  5E5C 0000 0000 0060
02DC  5E5C 0000 0000 0060
02D8  5E5C 0000 0000 0060
02DA  5E5C 0000 0000 0060
02D8  5E5C 0000 0000 0060
02CE  5E5C 0000 0000 0060
02CB  5E5C 0000 0000 0060
02C8  5E5C 0000 0000 0060
02C5  5E5C 0000 0000 0060
02C2  5E5C 0000 0000 0060
02BA  2F5C 0000 0000 0060
0281  2F5C 0000 0000 0060
029C  2F3C 0000 0000 0060
0294  2F5C 0000 0000 0060
028F  2F5C 0000 0000 0060
028C  2F5C 0000 0000 0060
0289  2F5C 0000 0000 0060
0286  2F5C 0000 0000 0060
0283  2F5C 0000 0000 0060
0261  266C 0000 0000 0040
025E  261C 0000 0000 0040
0255  27CC 0000 0000 0040
024D  4EBC 0000 0000 0040
023D  2F3C 0000 0000 0060
023C  51CC 0000 0000 0060
0236  2F3C 0000 0000 0060
0230  51CC 0000 0000 0060
0216  216C 0000 0000 0040
0215  21F0 0000 0000 0040
0213  21BC 0000 0000 0040
020D  216C 0000 0000 0040
020C  21F0 0000 0000 0040
0209  21F0 0000 0000 0040
0208  216C 0000 0000 0040
0205  2090 0000 0000 0040
0203  219C 0000 0000 0040
01FB  2F5C 0000 0000 0060
01EA  1EA0 0000 0000 0040
01E9  1EFC 0000 0000 0040
01E1  66E0 0000 0000 0040
01DA  66E0 0000 0000 0040
01B5  1BA0 0000 0000 0040
019F  1A4C 0000 0000 0040
0161  27FC 0000 0000 0060
013C  144C 0000 0000 0060
0124  169C 0000 0000 0040
010C  144C 0000 0000 0060
00F7  0FB0 0000 0000 0040
00F2  0FBC 0000 0000 0040
00ED  0F7C 0000 0000 0040
00EC  0F2C 0000 0000 0040
00E9  0ED0 0000 0000 0040
0085  0B70 0000 0000 0040
0092  0960 0000 0000 0040
0077  0790 0000 0000 0060
006E  0790 0000 0000 0060
```

```
0C6C  0830 C0C0 003C 0060
0C66  081C 00C0 0C0C C060
0C5B  097C 00C0 0C3C 006C
0C5A  087C 00C3 0C3C C060
0C59  05BC C0C0 0C3C C040
0C51  5CFC 00C0 C030 0040
0C50  081C 00C0 0C0C C060
0C49  051C 00C0 0C0C 0060
0C46  05EC 00C3 C03C 0040
0C42  051C 0CC0 0C0C C060
0C41  0460 C3C0 0C30 0060
0C36  060C 0000 003C 004C
0C35  1540 0C00 0030 004C
0C32  1DDC C0C0 0C3C 0040
0C2F  181C 00C3 C03C 0040
0C2C  1BFC 00C0 CC30 0040
0C29  1AAC 0CC0 0C30 C04C
0C26  162C 00C0 0C0C 004C
0C23  1790 0CC0 0C0C C04C
0C20  036C C3C0 C030 C040
0C1D  0BF0 C000 0C00 C040
0C1A  048C 6000 0C0C 0040
0C14  0A8C 0000 C000 0040
0C11  0DDC 0000 0C3C C040
0C0A  046C 0C00 003C 0C60
0C07  06C0 0C00 C030 C040
 0 ERRORS    7714 CARDS.  36055 TOKENS.      65156 RULES.      5275 SECONDS.
 0 WARNINGS.      272 DISK SECTORS.
```

BIBLIOGRAPHY

1. Air Force Avionics Lab, Wright-Patterson AFB, Ohio, AFAL-TR-74-180, A Stack Machine Emulation, Anderson, C. M., January 1975.

2. Agrawala, A. K. and Rausher, T. G., "Microprogramming: Perspective and Status," IEEE Trans, C23, p. 817-37, August 1974.

3. Burkhard, W. A., "Flexible Computer Architectures for Research and Development," Computer Conference, Fall 1976.

4. Burroughs Corporation Report 64116, Emulation Facility, by Advanced Design Organization, Paoli, Pa., 28 June 1971.

5. Burroughs Corporation Report TR70-2, The Interpreter, by R. L. Davis, 16 February 1970.

6. Burroughs Corporation Report TR70-8, Microprogramming Manual for the Interpreter Based Systems, by Advanced Design Organization, Paoli, Pa., November 1970.

7. Burroughs Corporation Report 66143, Algol Reference Manual for the Interpreter Based System, by Advanced Design Organization, Paoli, Pa., 15 June 1975.

8. Computer Sciences Corporation, A Cost-Effective Emulation Approach for U. S. Army Telecommunications, by J. W. Carroll, p. 1-13.

9. Naval Electronics Laboratory Center and M. I. T. Sloan School of Management, Facilities Orientation Report, Volume 1, A Survey of the Navy Tactical Computer Applications and Executives, by Professors S. E. Madnick and J. D. Detreville, October 1975.

10. Haggerty, J. M. and Hartling, J. M., Emulation of the AN/UYK-7 Tactical Data Computer on the Burroughs D-Machine, Masters Thesis, Naval Postgraduate School, Monterey, California, December 1976.

11. Husson, S. S., Microprogramming: Principles and Practices, Prentice-Hall, Inc., 1970.

12. Jones, L., "Instruction Sequencing in Microprogrammed Computers," Proceedings NCC, 44, p. 91-8, 1975.

13. Lichstein, H. A., "When Should You Emulate?," Datamation, 15, p. 205-10, November 1969.

14. Mallach, E. G., "Emulator Architecture," Computer, p. 24-32, August 1975.

15. Mercer, R. J., "Microprogramming," Journal for the Association of Computing Machines, 4, p. 157-71, April 1957.

16. Naval Material Command UNCLASSIFIED Letter MAT 09Y:JER Serial 130 to Naval Electronic Systems Command, Subject: Standard Shipboard Tactical Digital Processors and Program Languages, 29 May 1973.

17. Reigel, E. W., Faber, U., and Fisher, D. A., "The interpreter - A microprogramming building block system," Proceedings SJCC, 40, p. 705-23, 1972.

18. Rosin, R. F., "Contempory Concepts of Microprogramming and Emulation," Computer Surveys, 1, p. 197-212, December 1969.

19. Saal, H. J. and Schuster, L. J., On Measuring Computer Systems by Microprogramming.

20. Sperry Rand, UNIVAC, Computer Set AN/UYK20 Technical Manual Operations and Maintenance with Parts List, N 00039-73-C-0432, September 1974.

21. Sperry Rand, UNIVAC, AN/UYK-20 Computer Repertiore of Instructions, Fall 1976.

22. Sperry Rand, UNIVAC, AN/UYK-20 Technical Description, November 1976.

23. Sperry Rand, UNIVAC, User's Handbook for AN/UYK-20(V) Computer Volume 3 (Change 4), NAVELEX 0967-LP-598-2030, April 1976.

24. Wilkes, M. B., "The Growth of Interest in Microprogramming: A Literature Survey," Computer Surveys, 1, p. 139-45, September 1969.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexanderia, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
4.	LT Lyle V. Rich, SC, USN, Code 52Rs (Thesis Advisor) Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
5.	Asst Professor V. Michael Powers, Code 52Pw (Second Reader) Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
6.	Chief of Naval Material (Attn: Code 09Y) Department of the Navy Washington, D. C. 20360	1
7.	Mr. J. Lynch Burroughs ADD Federal and Special Systems Group P. O. Box 517 Paoli, Pennsylvania 19301	1
8.	Mr. C. Benson Naval Electronics Systems Engineering Center P. O. Box 37 San Diego, California 92138	1

9. Mr. J. Locata 1
Burroughs Corporation
P. O. Box 517
Paoli, Pennsylvania 19301

10. Mr. J. Westergren 1
Field Engineer
Univac Computer Systems
P. O. Box 3525
St. Paul, Minnesota 55165

11. CAPT Ralph H. Anzelmo, USMC 1
15767 Edgewood Drive
Montclair Country Club Lake
Dumfries, Virginia 22026

12. LT Theodore L. Kaye, USN 1
3880 Fairview Road
Reno, Nevada 89511

Thesis 169582
A567 Anzelmo
c.1 Emulation of the AN/UYK-20 tactical data computer on the Burroughs D-machine.